

FPT UNIVERSITY

Capstone Project Document

Mobile Tracking System

Mobile Tracking System		
Group Members	Vũ Thúy Nhâm	00875
	Nguyễn Trung Kiên	00801
	Vũ Hoàng Giang	00848
	Trần Phương Thảo	00943
	Phạm Tùng Bách	01389
Supervisor	Mr. Nguyen Tat Trung	
Capstone Project code	MTS	

Record of Changes

Date	Change Item	Description	By	Version
10/09/2012		Create the document	BachPT	0.1
17/09/2012	Proposal	Update figure	NhamVT ,ThaoTP	0.2
21/09/2012	Software Project Plan	Add new	NhamVT, BachPT	0.3
18/10/2012	Software Design	Add new	ThaoTP	0.4
05/12/2012	Testing	Add new	BachPT	0.5
20/12/2012	All	Update comment	NhamVT	0.6

Table of Contents

Table of Contents.....	1
Definitions and Acronyms.....	14
I. Chapter I - Project Introduction.....	15
1.1. Project Information.....	15
1.2. Purposes.....	15
1.3. Background	15
1.4. Literature Review of Existing Systems	16
1.5. Our Proposal	17
By implementing a helpful tool, our product will have the following advantage:	17
For smart phone, our product will have core functions below:	17
For web base system, our product will have core functions below:.....	17
Functions Proposal.....	18
Figure 3- Overall of RTS.....	19
II. Chapter II - Software Project Management Plan	19
2.1 Problem Identify	19
2.1.1 Core Business Case and Approach	19

2.1.2	Project milestones and deliverables	19
2.1.3	Project Standard Objectives.....	20
2.2	Project Organize.....	21
2.2.1	Software Process Model	21
Figure 4 – Iterative Software Process Model.....		21
2.2.2	Role and Responsibilities	21
Figure 5 - Project Team Members		22
2.3	Project Management Plan	22
Figure 6 - Project Management Plan		22
2.4	Hardware and Software	23
2.4.1	Hardware requirement	23
2.4.2	Software requirement.....	23
2.5	Success Criteria	23
2.6	Risk Management	24
2.7	Coding Convention for Java	25
2.7.1	File Name	25
2.7.2	File Organization	26
2.7.3	Indentation.....	26
2.7.4	Declarations	27
Number per Line		27
Placement		27
Initialization		28
Class and Interface Declarations		28
2.7.5	Statements.....	29
Simple Statements		29
Compound Statements		29
Return Statements		29
2.7.6	White Space	32
Blank Lines		32
Blank Spaces		32
2.7.7	Naming Conventions.....	33
III.	Chapter III - Software Requirement Specification	35

3.1. Communication protocol	35
3.2. High Level Requirements	35
3.1.1 Domain Model	35
Figure 7 – MTS Domain Diagram	35
Domain Object Description	36
3.1.2 User Case Diagram	37
Figure 8 - MTS Use Case Diagram	37
Use case description	37
3.1.3 Business Process Flow.....	38
Figure 9 - Business Process Flow.....	39
Business Flow List	39
3.2 Functional Requirement	40
3.2.1 Function “Position Saving”	40
Position data:	40
JSON Data Standard:	40
JSON Data Sample:	40
Detail Process:	41
Activity Diagram	42
Figure 10 - Position Saving Diagram	42
3.2.2 Function “Sms& Call Logs Saving”	44
SMS Data:	44
JSON SMS Data Standard:	44
JSON SMS Data Sample:	44
Call Logs Data:	44
JSON Call Logs Data Standard:	45
JSON Call Logs Data Sample:	45
Detail Process:	45
Activity Diagram	46
Figure 11 - Message and call logs saving	46
3.2.3 Function “Define Danger Area”	47
Danger Area data:	48
Define Danger Area Screen:	48

Figure 12 - Define Danger Area Screen.....	48
Detail Process:	48
Activity Diagram	50
Figure 13 - Define Danger Area Diagram	50
3.2.4 Function “Danger tracking”	51
Dangerous Areas Data:	51
JSON Data Standard:	51
JSON Data Sample:	51
Detail Process:	51
Activity Diagram:	53
Figure 14 - Danger Tracking Diagram.....	53
3.2.5 Function “Update configuration parameter”	54
Syntax of configuration sms: “ON”	54
Detail Process:	54
Activity Diagram:	55
Figure 15 - Update Configuration Parameter Diagram.....	55
3.2.6 Function “Display statistic information”	56
Position history screen:	56
Figure 16 - Position History Screen.....	57

SMS history screen:

SMS				
2012-12-07				
ID	Type	Phone Number	Time	Content
495	incoming	+84945709689	2012-12-07 01:56:42	Hay qua
496	incoming	+84945709689	2012-12-07 01:57:42	mtjg.atg.jatjg.ajg.ajg.atpg.akgajgajpg.ajtgaajamaja5m.agj
497	incoming	18001091	2012-12-07 10:07:07	(QC) Truy cap Facebook MIEN PHI trong 30 ngay tren u phi) de tai ung dung ngay! Tu choi QC, soan TC gui 180
498	incoming	+84946273186	2012-12-07 11:01:42	Gui cho a so cua Nham nhe.
499	outgoing	+84946273186	2012-12-07 11:43:03	Call mobile 01689914561
500	outgoing	+84946273186	2012-12-07 11:43:34	Ak a goi cho thang Tung beo chua

.....	57
Figure 17 - SMS History Screen	58
Figure 18 - SMS Information Detail.....	58
Call logs history screen:	58
Figure 19 - Call logs history screen	58
Figure 20 - Call Logs Information Detail.....	59
Detail Process:	59

Activity Diagram:

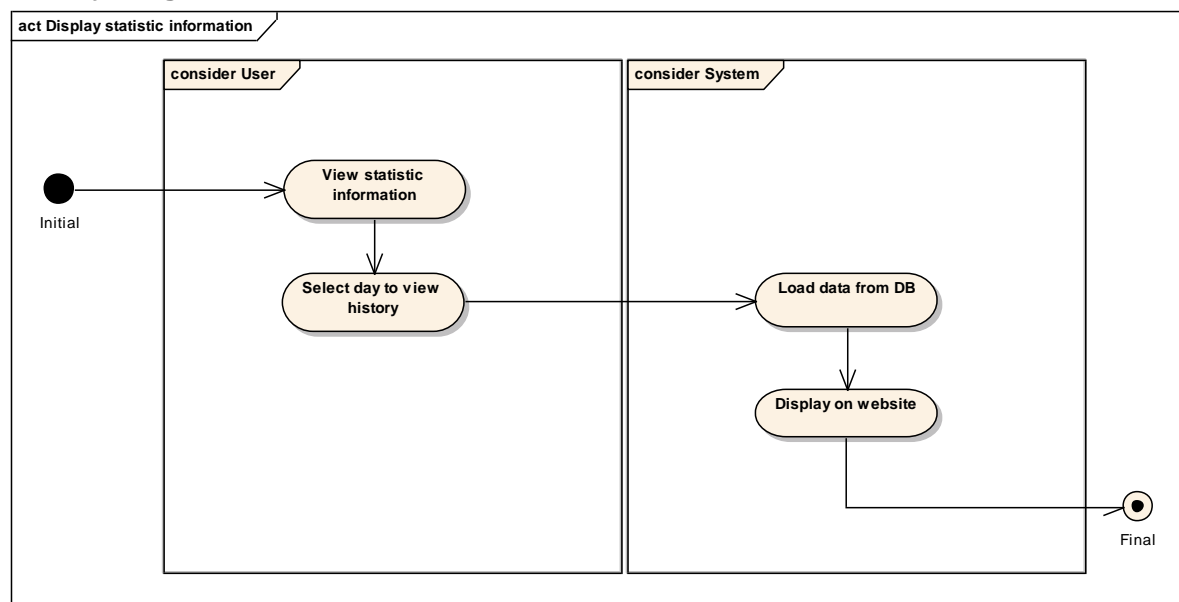


Figure 21 - Display statistic information diagram 60

3.2.7 Function "Register" 61

Register Screen: 61

Figure 22 - Register Screen 61

Registration Confirm Screen 61

Figure 23 - Registration Confirm Screen 62

Detail Process: 62

Activity Diagram 64

Figure 24 - Register Activity Diagram 64

3.2.8 Function "Login" 65

Login Screen 65

Figure 25 - Login screen 65

Detail Process 66

Activity Diagram 67

Figure 26 - Login Activity diagram 67

3.2.9 Function "Update Profile" 68

Update Profile Screen 68

Figure 27 - Update Profile Screen 68

Detail Process 69

Activities Diagram 70

Figure 28 - Update Profile Activity Diagram	70
3.2.10 Function “Manage Account”	71
Manage Account Screen:	72
Figure 29 - Update account screen	72
Figure 30 - Account Information Detail	72
Detail Process:	72
Activity Diagram	73
Figure 31 - Update Account Activity Diagram	74
3.3 Non – Functional Requirement	75
3.3.1 Usability.....	75
3.3.2 Reliability.....	75
Bug or Defect Rate	75
3.3.3 Availability.....	76
3.3.4 Security	76
3.3.5 Maintainability	76
3.3.6 Portability.....	77
3.3.7 Performance.....	77
• Response time under 10 seconds	77
IV. Chapter IV - Software Architecture Design	77
4.1 System Architecture Design	77
Figure 32 - System Overview	77
4.1.1 Mobile	78
Figure 33 - Mobile Architecture Design	78
4.1.2 Server	79
Figure 34 - Server Architecture Design	79
4.2 Class Diagram	80
4.2.1 Mobile	80
4.2.1.1 SMS	80
4.2.1.1.1 Class diagram	80
Figure 35 - SMS Class Diagram	80
4.2.1.1.2 Class explanation	80
4.2.1.1.2.1 SMSReceiver.....	80

4.2.1.1.2.2	SMSDataSource	81
4.2.1.1.2.3	SMSSentObserver	81
4.2.1.1.2.4	EnableConnectionService.....	82
4.2.1.1.2.5	DataBaseManager.....	83
4.2.1.1.2.6	SMSSentService.....	84
4.2.1.1.2.7	JSONParser	84
4.2.1.1.2.8	ConfigDataSource.....	85
4.2.1.1.2.9	GetParametersConfig.....	85
4.2.1.1.2.10	ConfigBean	86
4.2.1.1.2.11	SMSBean	86
4.2.1.1.2.12	GetMarkerConfig.....	87
4.2.1.2	Mobile Track	88
4.2.1.2.1	Class diagram	88
4.2.1.2.2	Class explanation	88
4.2.1.2.2.1	NetworkStateBroadcastReceiver	88
4.2.1.2.2.2	BootUpReceiver	89
4.2.1.2.2.3	PreferenceListener	89
4.2.1.2.2.4	SMSSentService.....	90
4.2.1.2.2.5	MobileTrackService.....	90
4.2.1.2.2.6	LocationTrackingListener	91
4.2.1.2.2.7	TrackerProvider.....	92
4.2.1.2.2.8	TrackerEntry	93
4.2.1.2.2.9	TrackDataHelper	94
4.2.1.2.2.10	EnableConnectionService.....	95
4.2.1.2.2.11	DataBaseManager.....	96
4.2.1.3	Call Logs.....	98
4.2.1.3.1	Class diagram	98
4.2.1.3.2	Class explanation	98
4.2.1.3.2.1	CallLogBean	98
4.2.1.3.2.2	CallLogObserver	99
4.2.1.3.2.3	CallLogDataSource	99
4.2.1.3.2.4	DataBaseManager.....	100

4.2.1.3.2.5	CallLogReceiver	102
4.2.1.4	Upload Service	102
4.2.1.4.1	Class Diagram	102
4.2.1.4.2	Class Explanation.....	103
4.2.1.4.2.1	EnableConnectionService.....	103
4.2.1.4.2.2	UploadDataService.....	103
4.2.1.4.2.3	UploadTracking	104
4.2.1.4.2.4	CSVWriter	104
4.2.1.4.2.5	JSONParser	105
4.2.1.4.2.6	DatabaseManager.....	106
4.2.1.4.2.7	CallLogBean.....	107
4.2.1.4.2.8	TrackingBean.....	107
4.2.1.4.2.9	AlarmManager	108
4.2.2	Server	109
4.2.2.1	Account	109
4.2.2.1.1	Class diagram	109
4.2.2.1.2	Class explanation	109
4.2.2.1.2.1	ConfirmSite	109
4.2.2.1.2.2	AccountDBFuntion	110
4.2.2.1.2.3	CheckLogin	110
4.2.2.1.2.4	ChangePasswordDB	111
4.2.2.1.2.5	UpdateAccountDB.....	111
4.2.2.1.2.6	UpdateAccount	111
4.2.2.1.2.7	Phpsqlajax_dbinfo.....	112
4.2.2.2	Danger Area	112
4.2.2.2.1	Class diagram	112
4.2.2.2.2	Class explanation	113
4.2.2.2.2.1	GenParameter.....	113
4.2.2.2.2.2	Phpsqlajax_genxmlMarkers	113
4.2.2.2.2.3	phpsqlinfo_addrow	113
4.2.2.2.2.4	phpsqlajax_dbinfo.....	114
4.2.2.3	Service	115

4.2.2.3.1	Class diagram	115
4.2.2.3.2	Class explanation	115
4.2.2.3.2.1	Handle_upload	115
4.2.2.3.2.2	DownLoadParameter	115
4.2.2.3.2.3	phpsqlajax_dbinfo.....	116
4.2.2.3.2.4	ImportCSV	116
4.2.2.4	Show History	117
4.2.2.4.1	Class diagram	117
4.2.2.4.2	Class explanation	117
4.2.2.4.2.1	Phpsqlajax_genxml3	117
4.2.2.4.2.2	Phpsqlajax_genxmlAllLocation.....	118
4.2.2.4.2.3	GenSms	118
4.2.2.4.2.4	GenCallLog	118
4.2.2.4.2.5	phpsqlajax_dbinfo.....	119
4.3	Sequence Diagram	119
4.3.1	Mobile	119
4.3.1.1	SMS	119
4.3.1.2	MobileTrack	121
4.3.1.3	Call Logs.....	122
4.3.1.4	UploadData	123
4.3.1.5	Danger Track	125
4.3.2	Server	126
4.3.2.1	Danger Area Sequence.....	126
4.3.2.2	Login.....	127
4.3.2.3	Register	128
4.3.2.4	Update.....	129
4.3.2.5	Show History	130
4.3.2.6	Download/Upload.....	131
4.4	Database	132
4.4.1	Mobile	132
4.4.2	Server	134
V.	Chapter V - Testing.....	135

5.1	Requirements for Test	135
5.1.1	Feature to be tested	135
5.1.1.1	Application	135
5.1.1.2	Website	136
5.1.2	Feature not to be tested	137
5.2	Test Strategy	137
5.2.1	Test Types	137
5.2.1.1	Function Testing.....	137
5.2.1.2	User Interface Testing.....	138
5.2.1.3	Performance Testing.....	139
5.2.2	Test Environment.....	140
5.3	Test cases	140
5.3.1	Application	140
5.3.1.1	Set Up.....	140
5.3.1.2	Running with 3G &GPS Are Turned On	141
5.3.1.3	Check Danger Areas	143
5.3.1.4	Check Receive Active Message	144
5.3.2	Central Web with Chrome	145
5.3.2.1	Login Screen	145
5.3.2.2	Register Screen	146
5.3.2.3	Update Account Screen	150
5.3.2.4	Danger Area Screen	155
5.3.2.5	History Screen	157
5.3.2.6	Admin Screen	160
5.4	Test Summary Report	163
5.4.1	Test Execution Summary.....	163
5.4.2	Test Results	163
5.4.2.1	Summary of Incident Reports Raised During Project.....	163
5.4.3	Conclusion.....	164
5.4.3.1	System Test Success Criteria	Error! Bookmark not defined.
5.4.3.2	Conclusion.....	Error! Bookmark not defined.
VI.	Software User's Manual.....	164

6.1	Installation Guide	164
6.1.1	Website	164
6.1.1.1	Setting Up the Environment at Server Side	164
6.1.1.2	Deployment at Server Side	165
6.1.1.3	Setting Up the Environment at Client Side	166
6.1.2	Application	166
6.1.2.1	Build and Get apk File	166
6.1.2.2	Set Up Application to Device	176
6.1.2.2.1	Prepare for Setting	176
6.1.2.2.2	Set Up Application.....	176
6.2	User's Guide	180
6.2.1	General Guide	180
6.2.1.1	Login.....	180
6.2.1.2	View Profile	181
6.2.1.3	View Position, SMS, Call Logs of Current Day	181
6.2.1.4	View Danger Area Screen.....	182
6.2.2	Form Guide.....	182
6.2.2.1	Register a New Account	182
6.2.2.2	Forgot Password	183
6.2.2.3	User's Update Information	183
6.2.2.4	Delete Tracking, SMS and Call Logs Data	184
6.2.2.5	Change Password	184
6.2.2.6	View History of Position, SMS, Call Logs	185
6.2.2.7	View Information of Danger Point	185
6.2.2.8	Create a New Danger Point.....	186
6.2.2.9	Update Information or Delete Danger Point	186
6.2.2.10	Set Time Upload Data	187
6.2.3	Admin Guide	188
6.2.3.1	View All Information of Users	188
6.2.3.2	Update Information of Users or Delete Users	188
6.2.4	User Enables Update For Device by Sending Message	189
VII.	References	190

Definitions and Acronyms

Acronym	Definition	Note
MTS	Mobile Tracking System	
GPS	Global Position System	
API	Application Programming Interface	
JSON	JavaScript Object Notation	
DB	Database	
BF	Business flow	
A	Activity	
UP	User process	
SP	System process	
SMS	Source Message Service	

Table 1 - Definitions and Acronyms

I. Chapter I - Project Introduction

1.1. Project Information

- Project name: **Mobile Tracking System**
- Project code: **MTS**
- Product type: **Android application and Web application**
- Timeline: **from September 2012 to December 2012**

1.2. Purposes

- The first purpose is to fulfill the requirements for Capstone Project.
- The second purpose we focus on creating a **complete application**, what has our dramatically desire to help users can locate mobile on the map and track people in real time.

1.3. Background

All people want to protect their relatives and keep them safe. And we are going to build mobile tracking application that helps them in securing life's most precious assets. Mobile tracking system designed specifically for relatives tracking will provide users the assurance that their relatives' location can be accessed at any time day or night.

For examples, your fiancée should be at work but she is not answering her phone and you're afraid she may have had an accident. With mobile tracking system you will know she's safe in the office. Your husband tells you he's going to be late tonight due to a business meeting. Is he really at the office, or is he having dinner somewhere else? Your kids tell you that they are staying late with friends to study for their test. Are they really there, or did they go somewhere else without letting you know?

You can now find out where they are, at any time, as long as using MTS with some features: locate mobile on map, track people in real time, alert when they are in danger...And your loved one will never know their mobile is being tracked because MTS remains completely hidden from the user. Once installed, you can secretly track their mobile activities 24/7 from your MTS user account.

1.4. Literature Review of Existing Systems

GPS Tracking Pro: is application that keeps users family and friends safe by tracking their real-time location using any Android phone. With the best in GPS technology, this app tracks the exact location of your family members and alerts you when they are in trouble, using the app's innovative check-in feature.

The application allows family members to locate each other on map.



Figure 1 - GPS Tracking Pro



Figure 2 - SMS Tracker Agent

SMS Tracker Agent is application that tracking phone and monitoring system for Android phones. Remotely track and monitor all SMS, MMS, text messages, calls, and GPS location. Users can remotely read all messages, photos sent and received, view phone call logs and view GPS location information on a map. All phone usage information can be remotely viewed on any web browser. Users now have a powerful tool to help keep your relatives safe even without physically see them.

Compare Existing Systems

Functions	GPS Tracking Pro	SMS Tracker Agent
Records movement location by GPS tracking	✓	✓
Alerts users when tracked mobile is in danger	✓	X
Saves messages and call logs	X	✓
Runs implicit	X	✓

Table 2 - Compare Existing Systems

1.5. Our Proposal

GPS Tracking Pro is application that tracks real time location to keep users' family and friends safe. This application allows family members to locate each other on map. It means not hidden by users, so doesn't run implicit and also not have saving mobile activities feature.

SMS Tracker Agent is tracking phone application. It allows remotely track and monitor all SMS, MMS, text messages, calls, and GPS location. Users can remotely read all messages, photos sent and received, view phone call logs and view GPS location information on a map. This app is phone tracking, so focus on tracking all mobile activities and hasn't developed alert when tracked phone is in danger feature yet.

By implementing a helpful tool, our product will have the following advantage:

- Application is installed as system application and cannot be removed in the usual way.
- Application is started automatically as soon as the phone is turned on.
- Application cans automatically active 3G, wifi of mobile.
- Application does not have a screen interface, runs fully implicit.
- End users can supervise most of mobile activities via web server anytime.

For smart phone, our product will have core functions below:

- Records movement locations.
- Saves sms and call logs.
- Synchronizes all information to website with internet access
- Alerts users in case their tracked mobile crosses the locations' boundaries or in dangerous areas

For web base system, our product will have core functions below:

- Allows register and login to system.
- Let's users secretly see tracking history of the tracked mobile (location, saved sms and call logs).
- Allows setting configuration parameters.
- Manage account of users.

Functions Proposal

Functions	GPS Tracking Pro	SMS Tracker Agent	MTS
Record movement location	✓	✓	✓
Save messages and call logs		✓	✓
Synchronize all information to website with internet access to remote tracking		✓	✓
Alert users in case their tracked mobile crosses the locations' boundaries or in dangerous areas	✓		✓
Setting configuration parameters		✓	✓
Run implicit to ensure confidentiality			✓
Show history of tracked mobile on web	✓		✓
Manage account of users			✓

Table 3 - Functions proposal

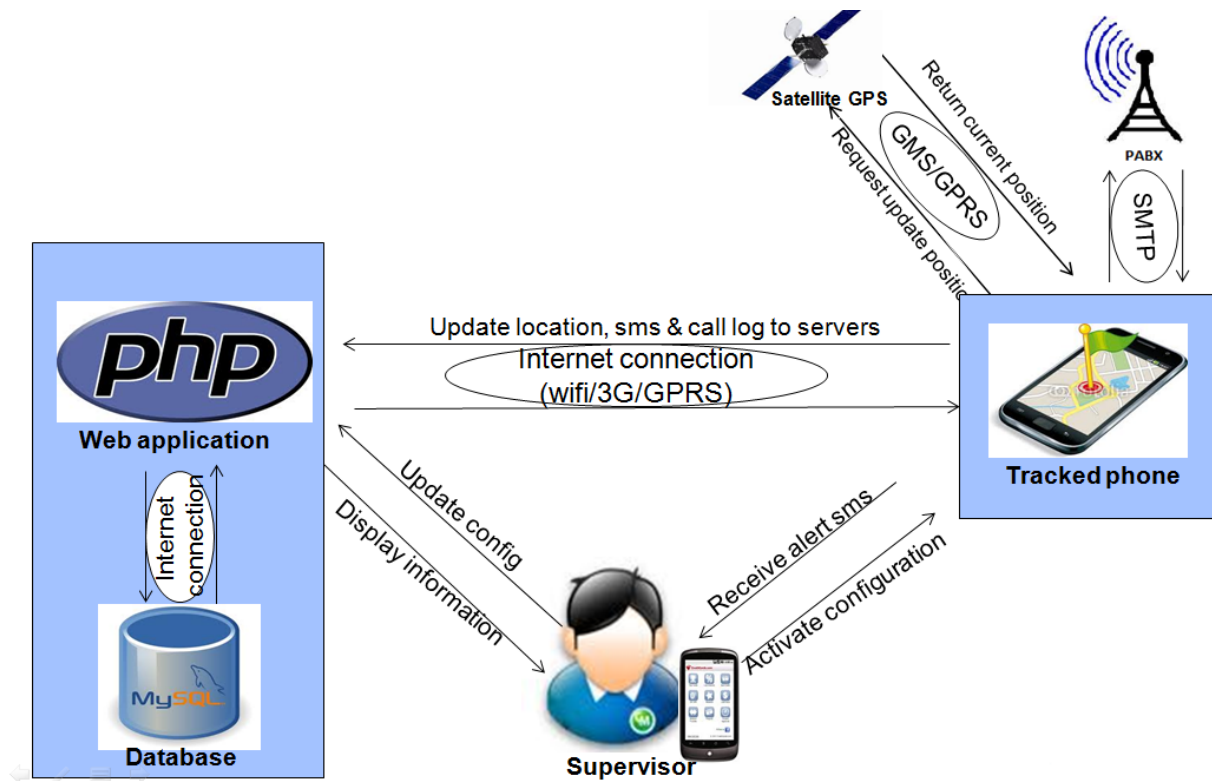


Figure 3- Overall of RTS

II. Chapter II - Software Project Management Plan

2.1 Problem Identify

2.1.1 Core Business Case and Approach

The system will provide the following main features in mobile application:

- Application is implicit software. It does not show its icon.
- System saves automatically real time position information from GPS system: longitude, latitude ...
- Record SMS
- Record call diary
- Application can upload all data to server.
- Users can customize the time for automatically upload data to server or the time for saving data on device.

2.1.2 Project milestones and deliverables

No	Stage	Committed	Actual	Deviation	Note
1	Report #1 - Project	17-Sep-12			

	Introduction				
2	Report #2 - Project Management Plan	24-Sep-12			
3	Report #3 - SRS & prototype V1.0	08-Oct-12			
4	Report #4 - Detailed design V1.0	22-Oct-12			
5	Final package	26-Nov-12			
6	Report #5 – Software Test Document	03-Dec-12			
7	User Manual V1.0	05-Dec-12			

Table 4 - Project Milestones and Deliverables

2.1.3 Project Standard Objectives

Metrics	Unit	Committed	Actual	Deviation
Start Date	dd-mmm-yy	05-Sep-12		
End Date	dd-mmm-yy	22-Dec-12		
Duration				
Maximum Team Size	Person	5		

Table 5 - Project Standard Objectives

2.2 Project Organize

2.2.1 Software Process Model

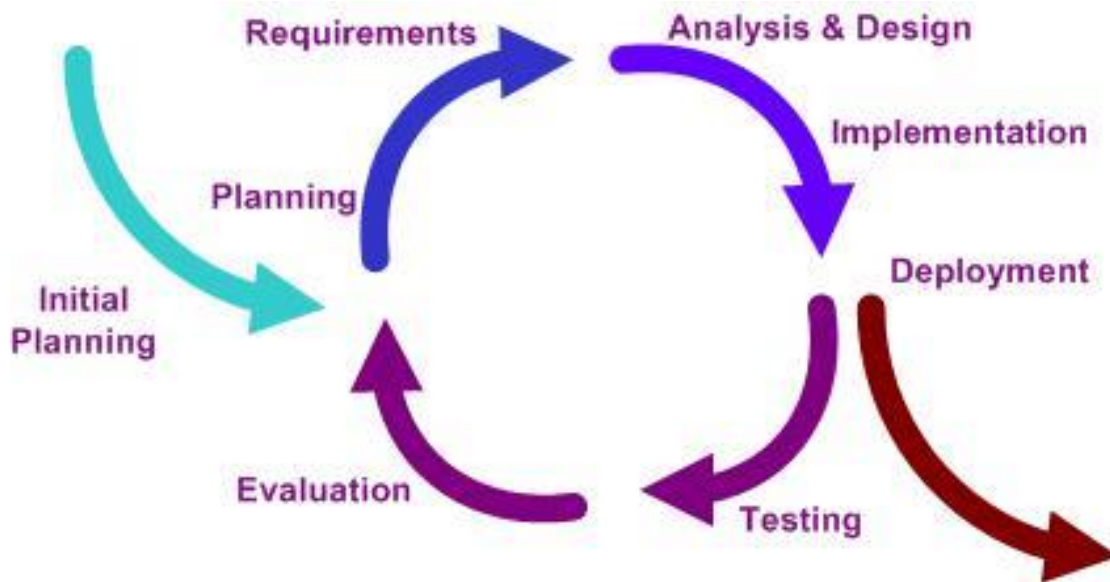


Figure 4 – Iterative Software Process Model

- ✓ In iterative model we can only create a high-level design of the application before we actually begin to build the product and define the design solution for the entire product. Later on we can design and build a skeleton version of that, and then evolved the design based on what had been built.
- ✓ In iterative model we are building and improving the product step by step. Hence we can track the defects at early stages. This avoids the downward flow of the defects.
- ✓ In iterative model we can get the reliable user feedback. When presenting sketches and blueprints of the product to users for their feedback, we are effectively asking them to imagine how the product will work.
- ✓ In iterative model less time is spent on documenting and more time is given for designing.

2.2.2 Role and Responsibilities

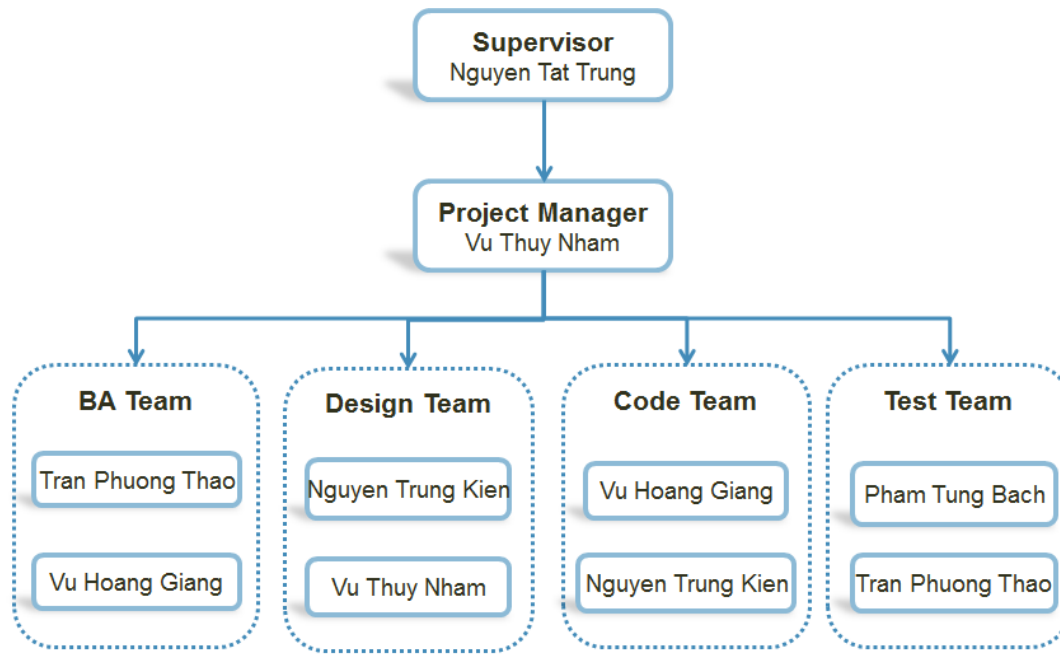


Figure 5 - Project Team Members

2.3 Project Management Plan

Refer to [MST Master plan.mpp](#) for more detail

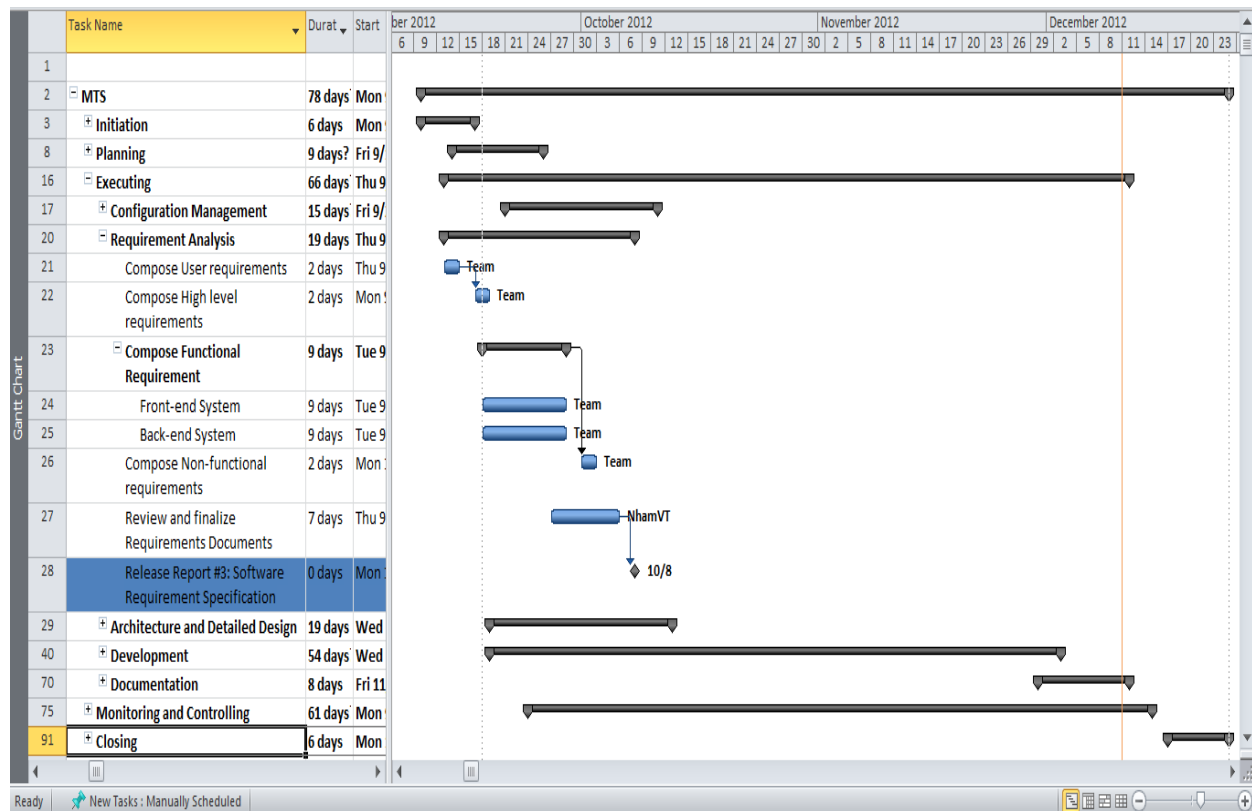


Figure 6 - Project Management Plan

2.4 Hardware and Software

2.4.1 Hardware requirement

- Smart phone with Android 2.3.3 operating system
- Server with minimum configuration: 4GB of RAM, 100GB of hard disk, Core 2 Duo

2.4.2 Software requirement

- Operating System Client: Android 2.3.3
- Operating System Server: Windows 7
- IDE: Eclipse and phpDesigner v8.0
- Development Language: PHP v7.0 and Java for Android
- DBMS: mySql

2.5 Success Criteria

At the completion of project, we expect to achieve the following criteria:

- ❖ Perspective on successful product
 - Good-looking and friendly interface with smart and simple design
 - Good performance
 - Value and stable
 - Convenience and easy to set up
 - Good server for backup data
- ❖ Perspective on learning
 - Understanding the primary of process of making software
 - Apply this process for capstone project.
 - Research and apply technologies what have taught at university. For example: JAVA, SMARTDRAW, ENTERPRISE ARCHITECT, MYSQL...
 - Learn how to use new technologies, for example: JAVA ON ANDROID, PHP, and JSON...
 - Apply technologies to professional process what has suggested from masters (supervisor).
 - Training process of project management with team leader. Leader always store knowledge for getting necessary skills. She or he can use tools for management control.
 - Capture the importance of project quality management and can use tools, techniques for quality control.
 - Designer gets keys for making friendly, smart and simple interfaces.
- ❖ Perspective on project management

- Keep deadline
- Achieve all quality requirement without overtime
- Improve cohesion of members in team

2.6 Risk Management

ID	Risk Description	Avoidance	Contingency Plan	Status
1	Poor experience of management makes plan late.	There is always a group with 2 or three members, for example: 2 main coders Kien and Giang, 3 remain members Bach, Thao, Nham. All groups will often meet least twice a week. Leader will update result of each meeting. Leader need to take care about team member, and find some strength, weakness ... Learn more about process and how to assign jobs.	All team members will have meeting for searching solution to help leader.	Closed
2	Poor experience of evaluation make miss scope.	List tasks and check continuously. Evaluate quality and progress weekly.	We overtime if a task is going miss deadline	Closed
3	Bad communication breakdown can make changing time, work and delay plan	We need using words more clearly, talk with each other more, note and send email to confirm information.	Maybe there are daily meeting for all team to make information clearly. If it necessary team need to meet some master.	Closed

4	Magnetic storms from sun can influence to GPS system. It creates large accuracy	We don't know when it happens, and we can't avoid that storm. But the time that storm impact to GPS is small time and rarely. We update information about position information frequently	We must update about position information always 1 second or smaller time.	Closed
5	Testing requests rooting device several time. It can damage device, lead to having no device for test	We check status of device continuously	We prepare two devices for risk.	Closed

Table 6 - Risk Management

2.7 Coding Convention for Java

2.7.1 File Name

Java Soft uses the following file suffixes:

File Type	Suffix
Java source	.java
Java byte code	.class

Frequently used file names include:

File Type	Use
GNUmakefile	The preferred name for makefiles. We use gnumake to build our software.
README	The preferred name for the file that summarizes the contents of a particular directory.

2.7.2 File Organization

Java source files have the following ordering:

- **Beginning comments**

All source files should begin with a c-style comment that lists the programmer(s), the date, a copyright notice, and also a brief description of the purpose of the program. For example:

```
/*  
 * Classname  
 *  
 * Version info  
 *  
 * Copyright notice  
 */
```

- **Package and Import statements**

The first non-comment line of most Java source files is a package statement. After that, import statements can follow. For example:

```
package java.awt;
```

```
import java.awt.peer.CanvasPeer;
```

2.7.3 Indentation

Four spaces should be used as the unit of indentation. The exact construction of the indentation (spaces vs. tabs) is unspecified. Tabs must be set exactly every 8 spaces (not 4).

- **Line Length**

Avoid lines longer than 80 characters, since they're not handled well by many terminals and tools.

Note: Examples for use in documentation should have a shorter line length—generally no more than 70 characters.

- **Wrapping Lines**

When an expression will not fit on a single line, break it according to these general principles:

- ✓ Break after a comma.
- ✓ Break before an operator.
- ✓ Prefer higher-level breaks to lower-level breaks.

- ✓ Align the new line with the beginning of the expression at the same level on the previous line.
- ✓ If the above rules lead to confusing code or to code that's squished up against the right margin, just indent 8 spaces instead.

2.7.4 Declarations

Number per Line

One declaration per line is recommended since it encourages commenting. In other words,

```
int level; // indentation level

int size; // size of table
```

is preferred over

```
int level, size;
```

In absolutely no case should variables and functions be declared on the same line. Example:

```
longdbaddr, getDbaddr(); // WRONG!
```

Do not put different types on the same line. Example:

```
int foo, fooarray[]; //WRONG!
```

Note: The examples above use one space between the type and the identifier. Another acceptable alternative is to use tabs, e.g.:

```
int level;      // indentation level
int size;      // size of table
Object currentEntry; // currently selected table entry
```

Placement

Put declarations only at the beginning of blocks. (A block is any code surrounded by curly braces "{" and ".") Don't wait to declare variables until their first use; it can confuse the unwary programmer and hamper code portability within the scope.

```
void MyMethod() {
int int1;      // beginning of method block
if (condition) {
int int2;    // beginning of "if" block
...
}
```

```
}
```

The one exception to the rule is indexes of for loops, which in Java can be declared in the for statement:

```
for (int i = 0; i < maxLoops; i++) { ...
```

Avoid local declarations that hide declarations at higher levels. For example, do not declare the same variable name in an inner block:

```
int count;
...
func() {
    if (condition) {
        int count;    // AVOID!
        ...
    }
    ...
}
```

Initialization

Try to initialize local variables where they're declared. The only reason not to initialize a variable where it's declared is if the initial value depends on some computation occurring first.

Class and Interface Declarations

When coding Java classes and interfaces, the following formatting rules should be followed:

- No space between a method name and the parenthesis "(" starting its parameter list
- Open brace "{" appears at the end of the same line as the declaration statement
- Closing brace "}" starts a line by itself indented to match its corresponding opening statement, except when it is a null statement the "}" should appear immediately after the "{"

```
class Sample extends Object {
    int ivar1;
    int ivar2;
    Sample(int i, int j) {
        ivar1 = i;
        ivar2 = j;
    }
}
```

```

    }
    intemptyMethod() {}
    ...
}

```

- Methods are separated by a blank line

2.7.5 Statements

Simple Statements

Each line should contain at most one statement. Example:

```
argv++; argc--; // AVOID!
```

Do not use the comma operator to group multiple statements unless it is for an obvious reason.

Example:

```

if (err) {
    Format.print(System.out, "error"), exit(1); //VERY WRONG! }

```

Compound Statements

Compound statements are statements that contain lists of statements enclosed in braces "{statements}". See the following sections for examples.

- The enclosed statements should be indented one more level than the compound statement.
- The opening brace should be at the end of the line that begins the compound statement; the closing brace should begin a line and be indented to the beginning of the compound statement.
- Braces are used around all statements, even singletons, when they are part of a control structure, such as an if-else or for statement. This makes it easier to add statements without accidentally introducing bugs due to forgetting to add braces.

Return Statements

A return statement with a value should not use parentheses unless they make the return value more obvious in some way. Example:

```

return;
returnmyDisk.size();

```

```
return (size ? size : defaultSize);
```

if, if-else, if-else-if-else Statements

The if-else class of statements should have the following form:

```
if (condition) {  
    statements;  
}  
if (condition) {  
    statements;  
} else {  
    statements;  
}  
if (condition) {  
    statements;  
} else if (condition) {  
    statements;  
} else if (condition) {  
    statements;  
}
```

Note: if statements always use braces {}. Avoid the following error-prone form:

```
if (condition) //AVOID! THIS OMITTS THE BRACES {}!  
    statement;
```

for Statements

A for statement should have the following form:

```
for (initialization; condition; update) {  
    statements;  
}
```

An empty for statement (one in which all the work is done in the initialization, condition, and update clauses) should have the following form:

```
for (initialization; condition; update);
```

When using the comma operator in the initialization or update clause of a for statement, avoid the complexity of using more than three variables. If needed, use separate statements before the for loop (for the initialization clause) or at the end of the loop (for the update clause).

while Statements

A while statement should have the following form:

```
while (condition) {  
statements;  
}
```

An empty while statement should have the following form:

```
while (condition);
```

do-while Statements

A do-while statement should have the following form:

```
do {  
statements;  
} while (condition);
```

switch Statements

A switch statement should have the following form:

```
switch (condition) {  
case ABC:  
statements;  
/* falls through */  
case DEF:  
statements;  
break;  
case XYZ:  
statements;  
break;  
default:  
statements;  
break;  
}
```


Every time a case falls through (doesn't include a break statement), add a comment where the break statement would normally be. This is shown in the preceding code example with the `/* falls through */` comment.

Every switch statement should include a default case. The break in the default case is redundant, but it prevents a fall-through error if later another case is added.

try-catch Statements

A try-catch statement should have the following format:

```
try {  
    statements;  
} catch (ExceptionClass e) {  
    statements;  
}
```

2.7.6 White Space

Blank Lines

Blank lines improve readability by setting off sections of code that are logically related.

Two blank lines should always be used in the following circumstances:

- Between sections of a source file
- Between class and interface definitions

One blank line should always be used in the following circumstances:

- Between methods
- Between the local variables in a method and its first statement
- Before a block (see section 5.1.1) or single-line comment
- Between logical sections inside a method to improve readability

Blank Spaces

Blank spaces should be used in the following circumstances:

- A keyword followed by a parenthesis should be separated by a space.

Example:

```
while (true) {
    ...
}
```

Note that a blank space should not be used between a method name and its opening parenthesis. This helps to distinguish keywords from method calls.

- A blank space should appear after commas in argument lists.
- All binary operators except `.` should be separated from their operands by spaces. Blank spaces should never separate unary operators such as unary minus, increment (`++`), and decrement (`--`) from their operands. Example:

```
a += c + d;

a = (a + b) / (c * d);

while (d++ = s++) {

    n++;

}

prints("size is " + foo + "\n");
```

- The expressions in a for statement should be separated by blank spaces.

Example:

```
for (expr1; expr2; expr3)
```

- Casts should be followed by a blank. Examples:

```
myMethod((byte) aNum, (Object) x);

myFunc((int) (cp + 5), ((int) (i + 3)) + 1);
```

2.7.7 Naming Conventions

Naming conventions make programs more understandable by making them easier to read. They can also give information about the function of the identifier—for example, whether it's a constant, package, or class—which can be helpful in understanding the code.

The conventions given in this section are high level. Further conventions are given at (to be determined).

Identifier Type	Rules for Naming	Example
Classes	Class names should be nouns, in mixed case with the first letter of each internal word capitalized. Try to keep your class names simple and descriptive. Use whole words—avoid acronyms and abbreviations (unless the abbreviation is much more widely used than the long form, such as URL or HTML).	class Raster; class ImageSprite;
Interfaces	Interface names should be capitalized like class names.	interface RasterDelegate; interface Storing;
Methods	Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized.	run(); runFast(); getBackground();
Variables	Except for variables, all instance, class, and class constants are in mixed case with a lower-case first letter. Internal words start with capital letters. Variable names should be short yet meaningful. The choice of a variable name should be mnemonic—that is, designed to indicate to the casual observer the intent of its use. One-character variable names should be avoided except for temporary “throwaway” variables. Common names for temporary variables are i, j, k, m, and n for integers; c, d, and e for characters.	int i; char *cp; float myWidth;
Constants	The names of variables declared class constants and of ANSI constants should be all uppercase with words separated by under-scores (“_”). (ANSI constants should be avoided, for ease of debugging.)	int MIN_WIDTH = 4; int MAX_WIDTH = 999; int GET_THE_CPU = 1;

III. Chapter III - Software Requirement Specification

3.1. Communication protocol

The system uses HTTP1.0 (or over) connection to communicate with server. When tracked phone start up, the application will be auto started and connected to the Internet to check with server if there is new configuration parameters which will be downloaded. For some other services such as end users change configuration parameters and send configuration message to tracked phone, application will request data from targeted server to update the most recent information.

3.2. High Level Requirements

3.1.1 Domain Model

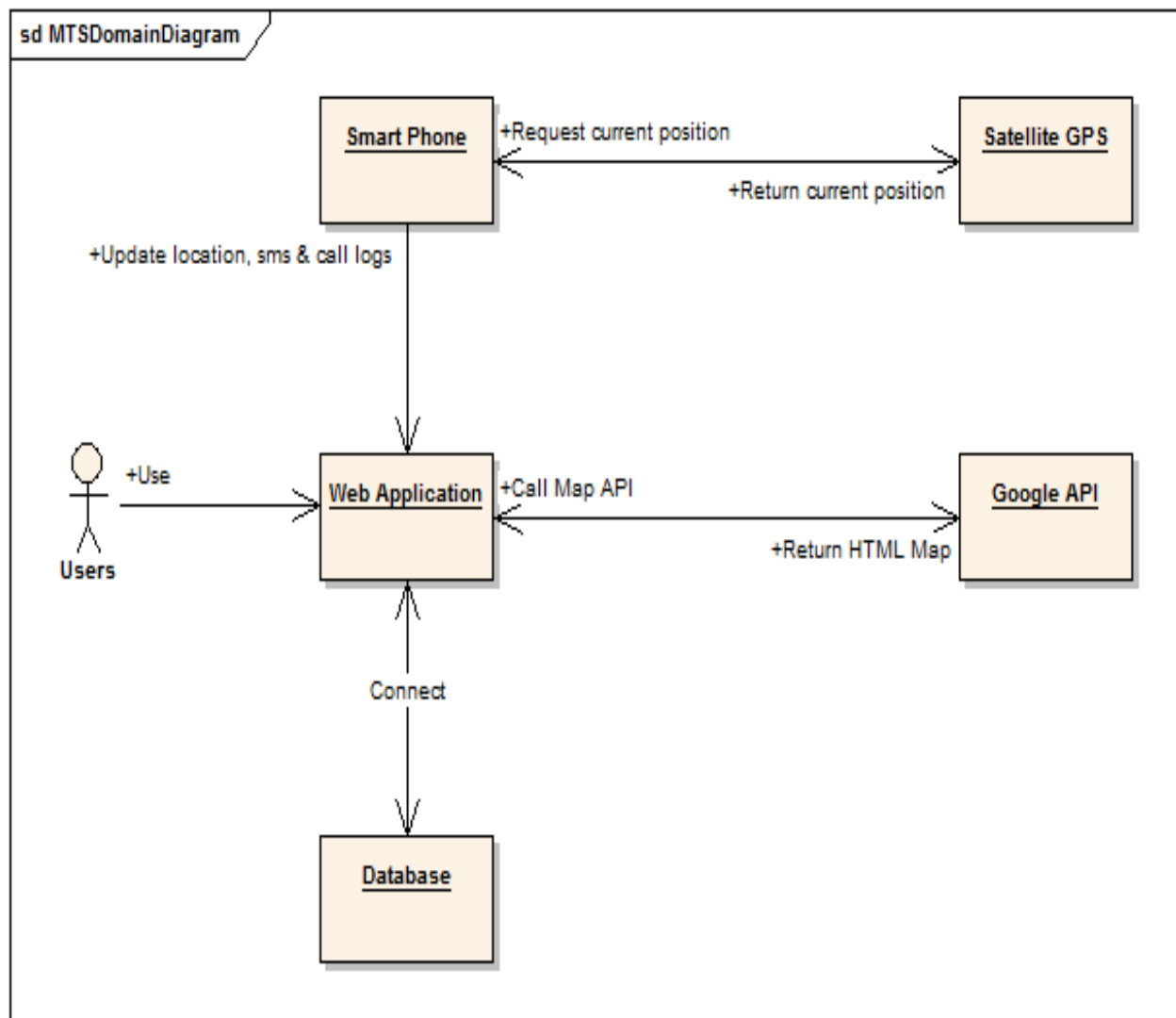


Figure 7 – MTS Domain Diagram

Domain Object Description

Object	Description
Smart Phone	A device which allows MTS application runs implicit. Users can set up application on this device to track without let smart phone user one knows.
Satellite GPS	A space-based satellite navigation system that provides location and time information in all weather, anywhere on or near the Earth, where there is an unobstructed line of sight to four or more GPS satellites.
Web application	Website is developed to allow users remotely tracking all movements of tracked phone. All movements are marked by lots of position on map. Users can also read all saved sms, call logs of tracked phone. By logging in website, users can set configuration to application. Follow the configuration setting, application is applied automatically when received configuration parameters.
Google API	A service support users view all information of tracked phone's positions on map. By using Google API service, web application can be developed few features to users easy to remote track.
Database	Database is built to store all tracked data include: positions, sms, call logs, dangerous areas information, configuration information...
Users	People who directly use MTS. Users set up application on smart phone. Then secretly remote tracking on website. Users can view all tracked positions, view all saved sms& call logs, send configuration parameter to application to let application run following to what and when users want.

Table 7 – Domain Diagram Description

3.1.2 User Case Diagram

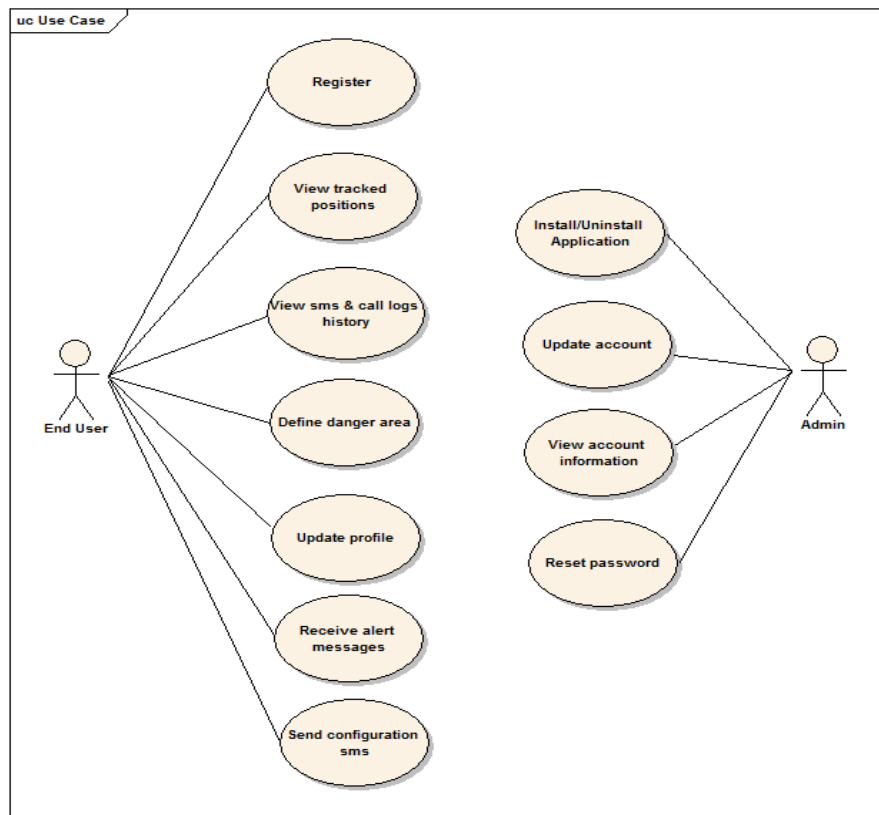


Figure 8 - MTS Use Case Diagram

Use case description

Role	Use case Name	Description
End User	Register	This use case allows users register to login to system. Each user must have an account to use system. All information of registered accounts will be stored in database. Then inform to users that successful registration and can login to system immediately.
End User	View tracked positions	This use case allows users track secretly positions on map when tracked phone is moving. When start tracking, after a fixed interval, the position is marked by draw a line following the position that has just passed. This process is fully secret.
End User	View sms& call logs history	This use case allows users view all sms and call logs of tracked phone. All smsand call logs of tracked phone

Role	Use case Name	Description
		are stored in temporary database. Then update to web server database after fixed times.
End User	Define danger area	This use case allows users define danger area on map via web server. Danger areas includes name of area, location, radius an comment about danger areas. Following list of danger areas, system can help users prevent their relatives when tracked mobile is in danger.
End User	Send configuration sms	This use case allows users send configuration sms to application any time. After receiving configuration sms with syntax "ON", application will be get automatically newest configuration parameter to apply.
End User	Receive alert messages	This use case allows users receive alert messages any time when tracked phone is in dangerous area.
End User	Update profile	This use case allows users update profile anytime via web server. Profile includes: User name, Imei, email, mobile.
Admin	Install/Uninstall Application	This use case allows admin install or uninstall application. Installing as admin role and cannot remove in usual way.
Admin	Update account	This use case allows admin update account such as: add new account, edit account or delete account.
Admin	View account information	This use case allows admin view information of selected account.
Admin	Reset password	This use case allows admin reset password when end user requests.

Table 8 – Use Case Description

3.1.3 Business Process Flow

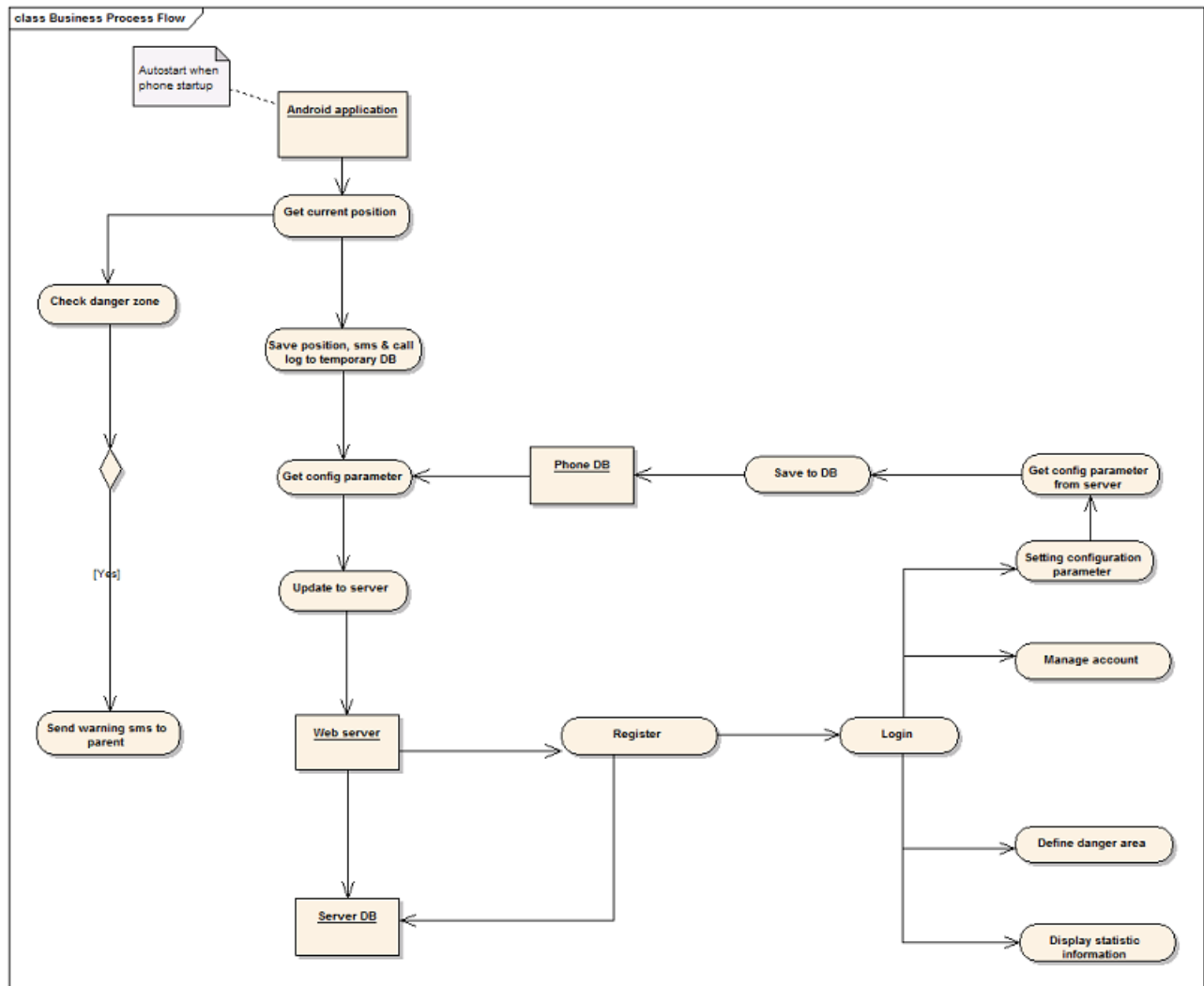


Figure 9 - Business Process Flow

Business Flow List

Business Flow Code	Name
BF1	Save Position, SMS & Call Logs

Business Flow Code	Name
BF2	Check Dangerous Areas
BF3	Setting configuration Parameters
BF4	Display Statistic Information
BF5	Register
BF6	Login
BF7	Manage Account

Table 9 – Business Flow List

3.2 Functional Requirement

3.2.1 Function “Position Saving”

In this function, system will use GPS to get current position of tracked phone, and then upload JSON object to server to synchronize.

Position data:

ID	Auto-incremented
IMEI	String
Time	DATETIME
Longitude	Long
Latitude	Long

Table 10 – Position Data Type

JSON Data Standard:

```
[{id,"imei","time", longitude, latitude}, {id,"imei","time", longitude, latitude}]
```

JSON Data Sample:

```
[{1,"012233423456098","2012-10-18T00:25:18Z", 21.0409, 105.7981},
{2,"012233423456098","2012-10-18T00:12:23Z", 34.0908, 102.2341}]
```

Detail Process:

Function Specification			
Function No	BF1A1	Function Version	1.0
Function Name	Position Saving		
Author			
Date	10/10/2012	Priority	Normal
Description System will use GPS to get current position of tracked phone, and then upload JSON object to server to synchronize.			
Inputs Longitude and latitude of position			
Source Longitude and latitude of position by Android API			
Outputs N/A			
Destination N/A			
Requires Network already turns on.(3G,Wifi)			
Pre-condition GPS is available.			
Post-condition Positions are stored in server database.			

Exception

In case be failed when upload to server database, system will:

- Logs exception.
- Check network to resend.

Table 11 – Position Saving Function Specification

Activity Diagram

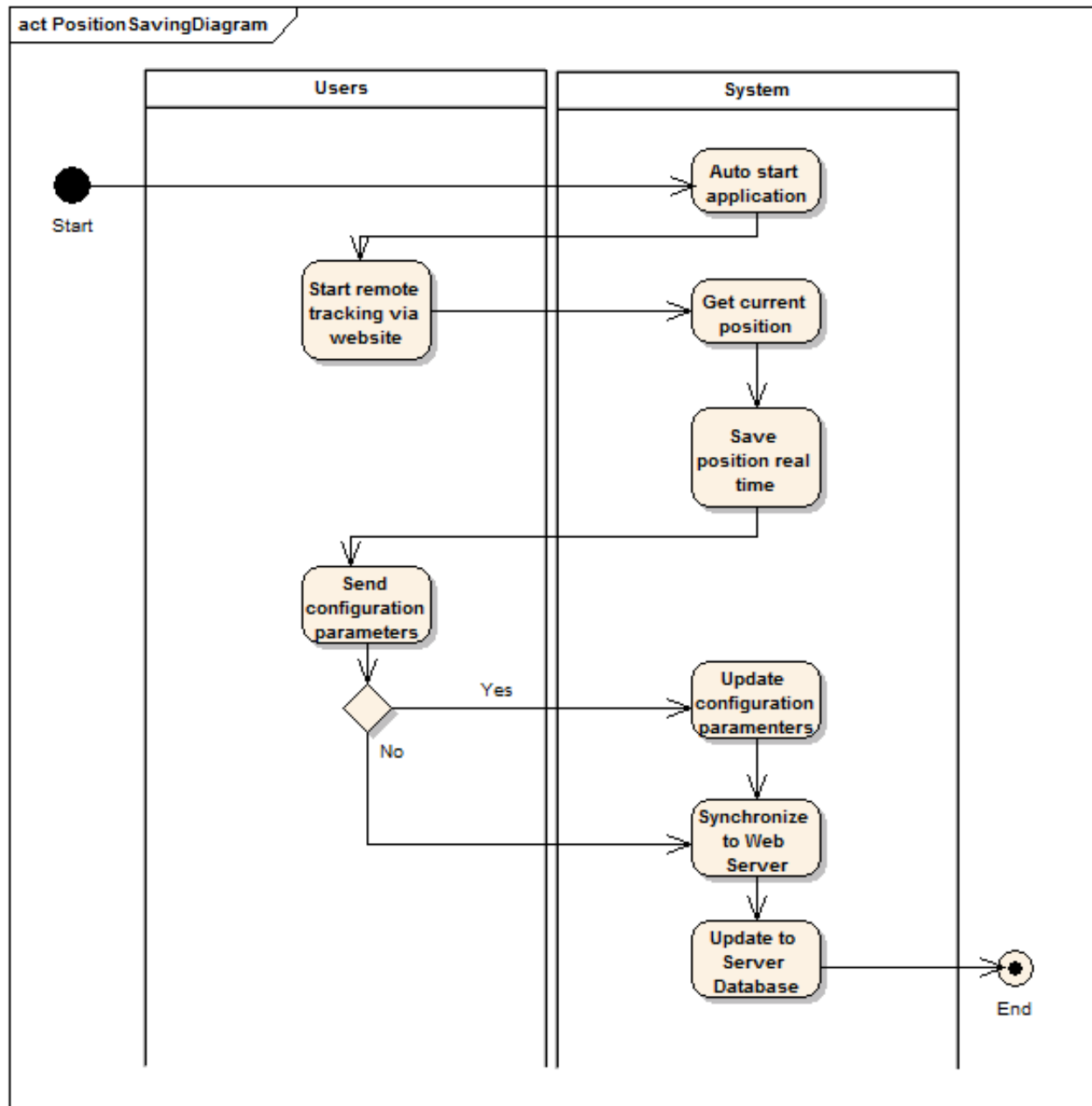


Figure 10 - Position Saving Diagram

Activity	Flow	Code	Description
Auto start application	BF1	A1.SP1	Application is set up on smart phone without let smart phone user one knows. So this application has to automatically start and run implicit. Application will automatically start immediately after starting phone. Since that, most of phone activities are tracked.
Start remote tracking via website	BF1	A1.UP2	When starting application, users can start remote tracking most phone activities such as: positions, messages and call logs. On website, users can track secretly position real time on map when tracked phone is moving. Users also can view all saved messages and call logs of tracked phone
Get current position	BF1	A1.SP3	By using GPS satellite, this application can get phone's current position. When moving, position is also updated.
Save position real time	BF1	A1.SP4	System saves positions real time when tracked phone is moving. When start moving, after a fixed interval, the position is saved.
Send configuration parameters	BF1	A1.UP5	Users permitted set configuration for application on web. Following this configuration, they can apply for application by sending a configuration parameter. Configuration maybe include: interval to update to server, phone number that sending alert messages to...
Update configuration parameters	BF1	A1.SP06	Before updating to server, system has to read newest configuration. If has a change, update this configuration immediately then continue to update.
Synchronize to web server	BF1	A1.SP7	After updating configuration, all just tracked positions are synchronized to web server. All tracked positions are displayed on map for users can be easy to view.
Update to server database	BF1	A1.SP8	After updating configuration, all just tracked positions are simultaneously updated to server database. These positions are saved as history. Users can view anytime later.

Table 12 – Position Saving Process Description

3.2.2 Function “Sms& Call Logs Saving”

In this function, system will save sms& call logs of tracked phone, and then upload JSON object to server to synchronize.

SMS Data:

ID	Auto-incremented
IMEI	String
Type	String
Phone Number	String
Time	DATETIME
Content	String

Table 13 –SMS Data Type

JSON SMS Data Standard:

```
[{"id","imei","type","phone_number","time","content"},  
{"id","imei","type","phone_number","time","content"},]
```

JSON SMS Data Sample:

```
[{"1","012233423456098","incoming","0982248952","2012-10-18T00:25:18Z","Hello"},  
{"2","012233423456098","outcoming","0982248952","2012-10-18T00:12:23Z","How are you"}]
```

Call Logs Data:

ID	Auto-incremented
IMEI	String
Type	String
Phone Number	String

Time	DATETIME
Duration	Time

Table 14 –Call Logs Data Type

JSON Call Logs Data Standard:

```
[{id, "imei", "type", "phone_number", "time", "duration"},
{id, "imei", "type", "phone_number", "time", "duration"}]
```

JSON Call Logs Data Sample:

```
[{"1","012233423456098","incoming","0982248952","2012-10-18T00:25:18Z","20:08"},
{"2","012233423456098","outcoming","0982248952","2012-10-18T00:12:23Z","10:09"}]
```

Detail Process:

Function Specification			
Function No	BF1A2	Function Version	1.0
Function Name	SMS & Call Logs Saving		
Author	MTS		
Date	10/10/2012	Priority	Normal
Description System will save sms & call logs of tracked phone, and then upload JSON object to server to synchronize.			
Inputs <ul style="list-style-type: none"> - Type, phone number, time and content of sms. - Type, phone number, time and duration of call. 			
Source <ul style="list-style-type: none"> - New sms: sent/received sms - New call: make/received call 			
Outputs			
Destination			
Requires			

Network already turns on.(3G,Wifi)
Pre-condition
Post-condition
Sms& call logs are stored in server database.
Exception
In case be failed when upload to server database, system will: <ul style="list-style-type: none"> - Logs exception. - Check network to resend.

Table 15 - Message & Call Logs Saving Function Specification

Activity Diagram

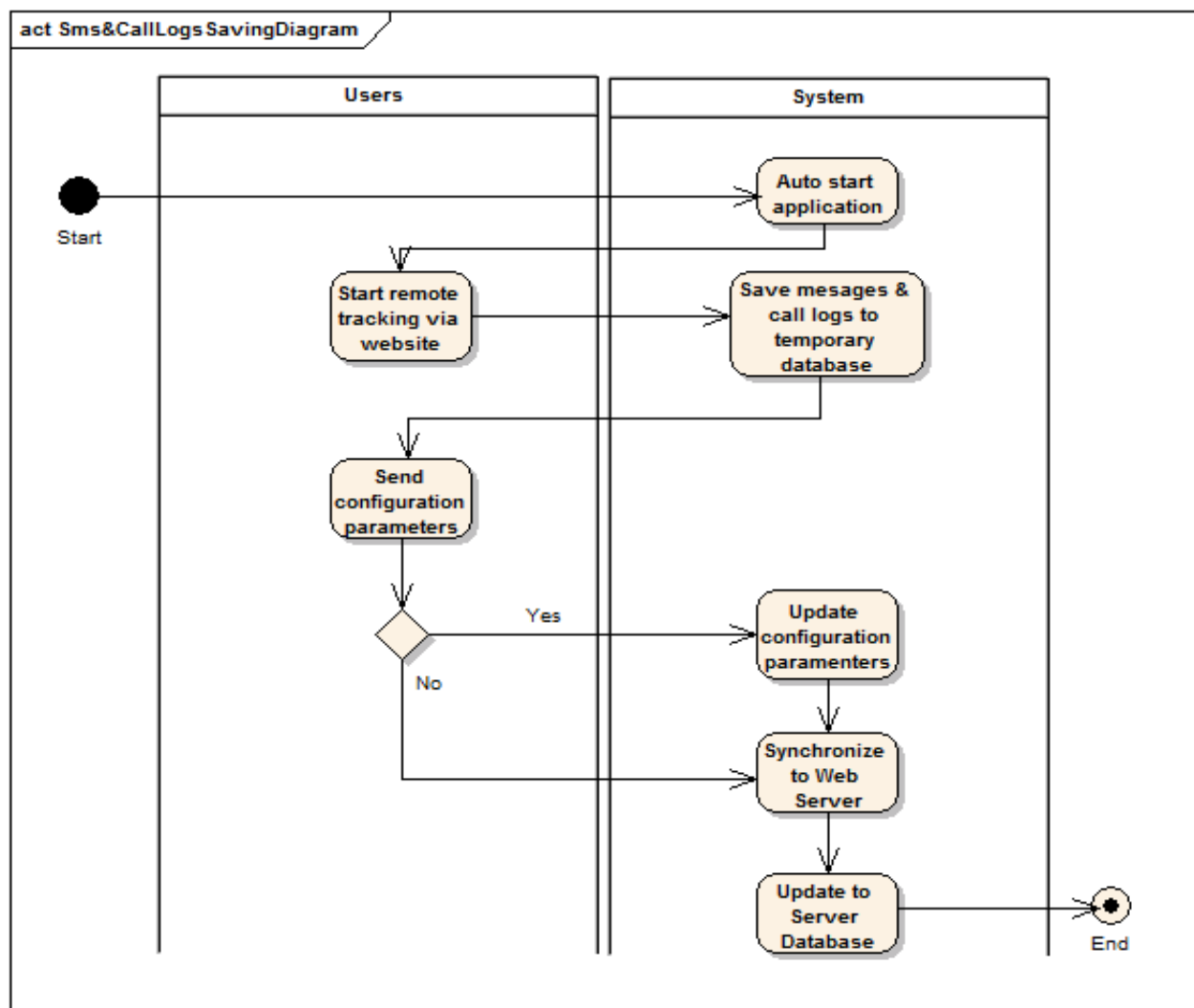


Figure 11 - Message and call logs saving

Activity	Flow	Code	Description
Auto start application	BF1	A2.SP1	Application is set up on smart phone without let smart phone user one knows. So this application has to automatically start and run implicit. Application will automatically start immediately after starting phone. Since that, most of phone activities are tracked.
Start remote tracking via website	BF1	A2.UP2	When starting application, users can start remote tracking most phone activities such as: positions, messages and call logs. On website, users can track secretly position real time on map when tracked phone is moving. Users also can view all saved messages and call logs of tracked phone
Save messages & call logs to temporary database	BF1	A2.SP3	System saves all messages & call logs of tracked phone. When tracked phone send/receive a messages or make call, this messages or call log is stored immediately in temporary database.
Send configuration parameters	BF1	A2.UP4	Users permitted set configuration for application on web. Following this configuration, they can apply for application by sending a configuration parameter. Configuration maybe include: interval to update to server, phone number that sending alert messages to...
Update configuration parameters	BF1	A2.SP5	Before updating to server, system has to read newest configuration. If has a change, update this configuration immediately then continue to update.
Synchronize to web server	BF1	A2.SP6	After updating configuration, all saved messages & call logs are synchronized to web server. And also are displayed to users view.
Update to server database	BF1	A2.SP7	After updating configuration, all saved messages & call logs are simultaneously updated to server database. These messages & call logs are saved as history. Users can view anytime later.

Table 16 SMS and Call Logs Saving Process Description

3.2.3 Function “Define Danger Area”

In this function, end users can click on map to define where danger area with their children is. Then system will save all about danger area to alert end user when your children go to there.

Danger Area data:

ID	Auto-incremented
IMEI	String
Longitude	Long
Latitude	Long
Radius	Long
Comment	String

Table 17 – Danger Area Data Type

Define Danger Area Screen:

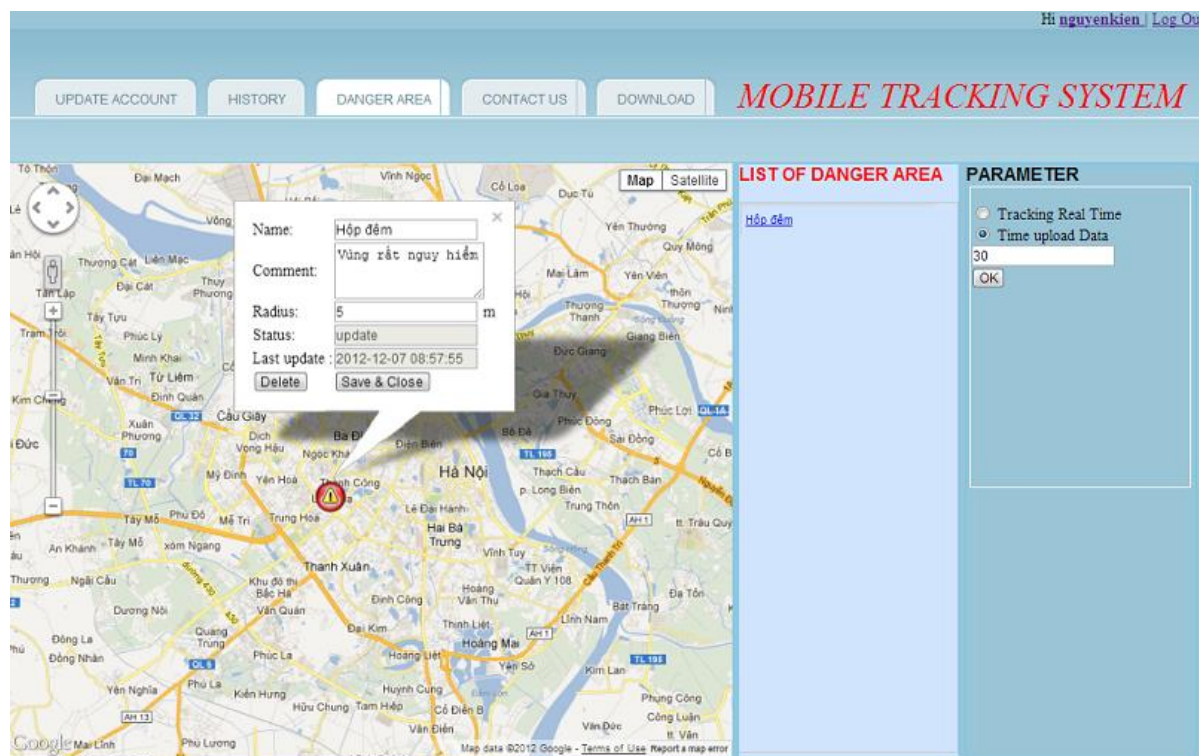


Figure 12 - Define Danger Area Screen

Detail Process:

Function Specification

Function No	BF2A1	Function Version	1.0
Function Name	Define Danger Area		
Author	MTS		
Date	10/10/2012	Priority	Normal
Description End users can click on map to define where danger area with their children is. Then system will save all about danger area to alert end user when your children go to there.			
Inputs End users input Name, Radius and Comment into Define Danger Area popup.			
Source Name, Radius and Comment from text fields of Define Danger Area popup.			
Outputs Danger marker is added to list markers and displayed on the left of map.			
Destination Map of Danger Area screen.			
Requires Danger Area screen is displayed. Map is loaded.			
Pre-condition End users login by users' role.			
Post-condition Sms & call logs are stored in server database.			
Exception In case be failed when save to server database, system will: <ul style="list-style-type: none"> - Logs exception. 			

Table 18 - Define Danger Area Function Specification

Activity Diagram

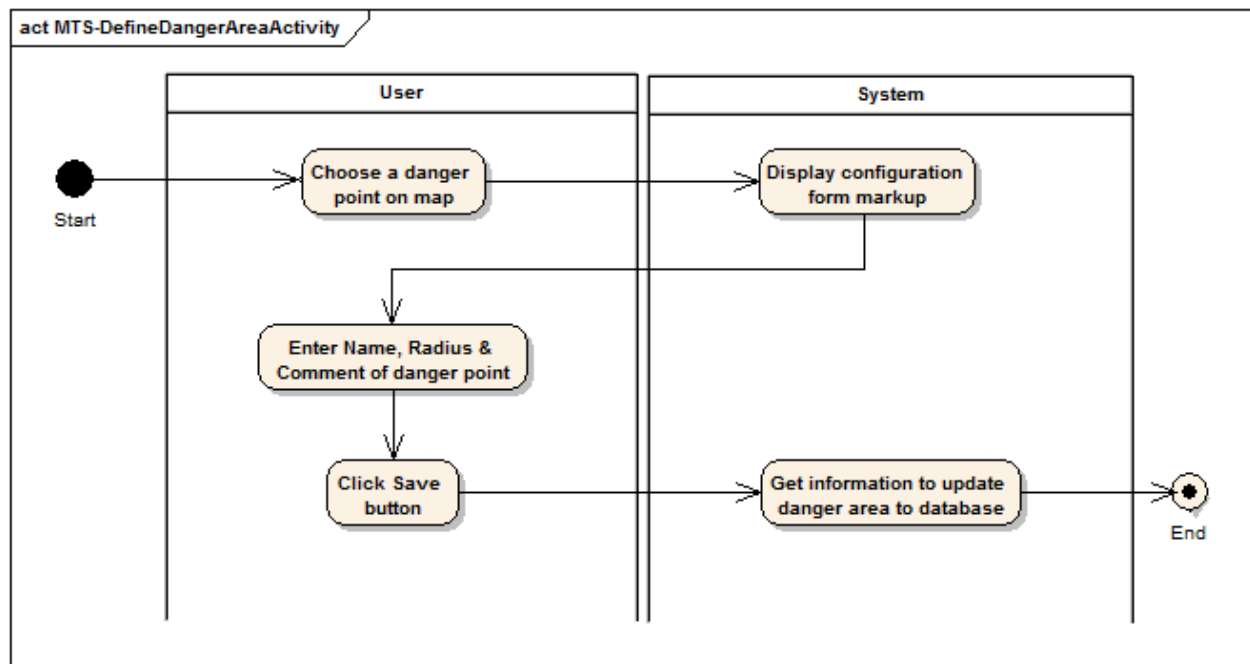


Figure 13 - Define Danger Area Diagram

Activity Name	Flow	Code	Description
Choose a danger point on map	BF2	A1.UP1	Danger area link is displayed in home screen. When user select danger area link, system will link to “Danger Area” screen and start “define danger area” flow. When map is loaded, users can pick a danger point on map.
Display configuration form popup	BF2	A1.SP2	After users pick a danger point on map, the configuration form popup is displayed, include: <ul style="list-style-type: none"> • Name: Name of danger area. • Radius: Distance between picked point and all point around. • Comment: Information of danger area.
Enter Name, Radius & Comment of danger point	BF2	A1.UP3	User must enter full of information about danger area, include: <ul style="list-style-type: none"> • Name: Name of danger area. • Radius: Distance between picked point and all point around. • Comment: Information of danger area.
Get information	BF2	A1.SP4	When users click on “Save” button, system executes updating new parameter to database. All

Activity Name	Flow	Code	Description
to update danger area to database			danger area information are updated immediately.

Table 19 - Define Danger Area Process Description

3.2.4 Function “Danger tracking”

In this function, according to list of dangerous areas, system will check and send alert messages to end user if tracked phone in danger.

Dangerous Areas Data:

ID	Auto-increment
IMEI	String
LONGITUDE	Long
LATITUDE	Long
RADIUS	Long

Table 20 - Dangerous Areas Data

JSON Data Standard:

[{id, "imei", longitude, latitude},{id, "imei", longitude, latitude}]

JSON Data Sample:

[{1,"012233423456098", 20.2345, 105.2345, 120}, {2,"012233423456098", 20.1223, 102.2398, 30}]

Detail Process:

Function Specification			
Function No	BF2A2	Function Version	1.0
Function Name	Danger Tracking		
Author	MTS		
Date	10/10/2012	Priority	Normal

Description
According to list of dangerous areas, system will check and send alert messages to end user if tracked phone in danger.
Inputs
Longitude and latitude of current position.
Source
Longitude and latitude of current position by Android API
Outputs
An alert message
Destination
Delivered
Requires
Network
Pre-condition
Network already turns on.(3G,Wifi)
Post-condition
GPS is available.
Exception

Table 21 - Danger Tracking Function Specification

Activity Diagram:

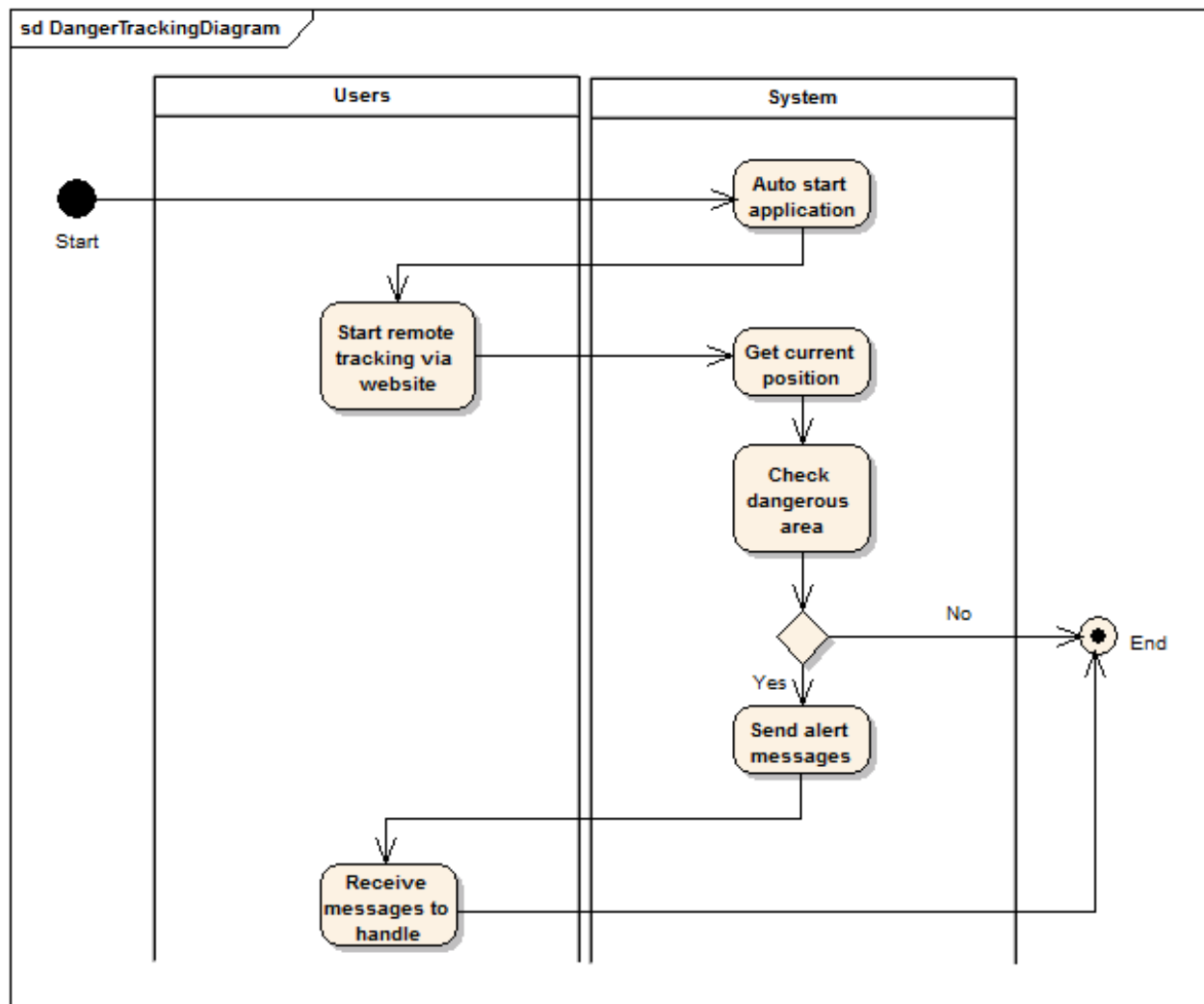


Figure 14 - Danger Tracking Diagram

Activity	Flow	Code	Description
Auto start application	BF2	A2.SP1	Application is set up on smart phone without let smart phone user one knows. So this application has to automatically start and run implicit. Application will automatically start immediately after starting phone. Since that, most of phone activities are tracked.
Start remote tracking via website	BF2	A2.UP2	When staring application, users can start remote tracking most phone activities such as: positions, messages and call logs. On website, users can track secretly position real time on map when tracked phone is moving. Users also can view all saved

Activity	Flow	Code	Description
			messages and call logs of tracked phone
Get current position	BF2	A2.SP3	By using GPS satellite, this application can get phone's current position. When moving, position is also updated.
Check dangerous area	BF2	A2.SP4	For each position, system will check dangerous following list of dangerous areas. List of dangerous areas are built by users. Users can define some dangerous areas. If tracked phone moves in danger, system returns true and call warning handling.
Send alert messages	BF2	A2.SP5	When tracked phone is in danger, system will immediately send alert messages to users. Messages may include dangerous location information.
Receive alert messages	BF2	A2.UP6	Users will receive alert messages immediately after system has just send. Receiving alert messages can help users have a best solution to protect relatives or own phone.

Table 22 - Danger Tracking Process Description

3.2.5 Function "Update configuration parameter"

In this function, when mobile receive configuration sms from end user, system will auto-update newest configuration parameter which end user has changed.

Syntax of configuration sms: "ON"

Detail Process:

Function Specification			
Function No	BF3A1	Function Version	1.0
Function Name	Update configuration parameter		
Author	MTS		
Date	10/10/2012	Priority	Normal
Description When mobile receive configuration sms from end user, system will auto-update newest configuration parameter which end user has changed.			
Inputs			

Configuration message with syntax "ON"
Source
Outputs
Destination
Requires
Mobile is available.
Pre-condition
Hide configuration message.
Post-condition
<ul style="list-style-type: none"> - Automatically turn on network. - Get configuration parameter from server immediately.
Exception
<p>In case be failed when get configuration parameter from server database, system will:</p> <ul style="list-style-type: none"> - Logs exception. - Configuration parameter of database is not changed.

Table 23 - Update Configuration Parameter Function Specification

Activity Diagram:

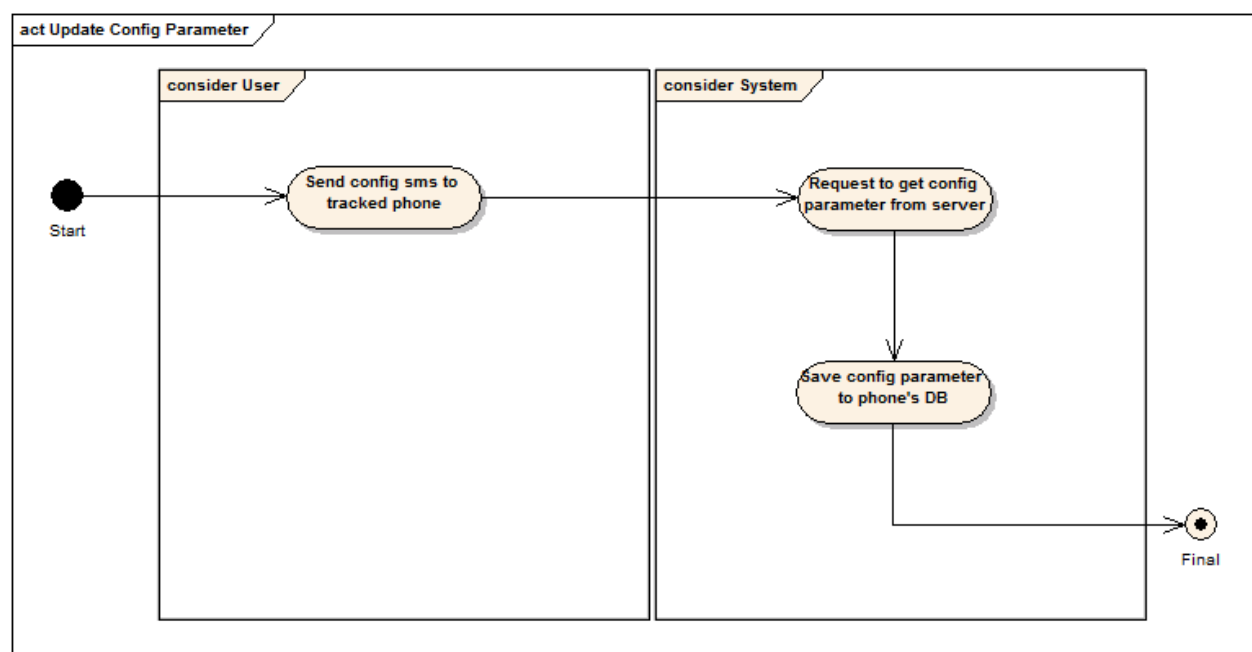


Figure 15 - Update Configuration Parameter Diagram

Activity Name	Flow	Code	Description
Send configuration sms to tracked phone	BF3	A1.UP1	To active new configuration parameters, user needs to send configuration sms to tracked phone. Content of configuration sms comply with following syntax: "ON"
Request to get configuration parameter from server	BF3	A1.SP2	After receive sms with defined syntax, MTS android application execute following steps: <ul style="list-style-type: none"> ❖ Hide configuration sms with phone user ❖ Auto connect network ❖ Send request to get new configuration parameters to server
Save configuration parameter to phone's DB	BF3	A1.SP3	Save data which recently get from server to database of tracked phone

Table 24 - Update Configuration Parameter Process Description

3.2.6 Function "Display statistic information"

In this function, system will display all tracking information for user. Displayed information is divided by date, includes:

- Position history
- SMS history
- Call logs history

<Refer data sample table in Appendix for more detail>

Position history screen:

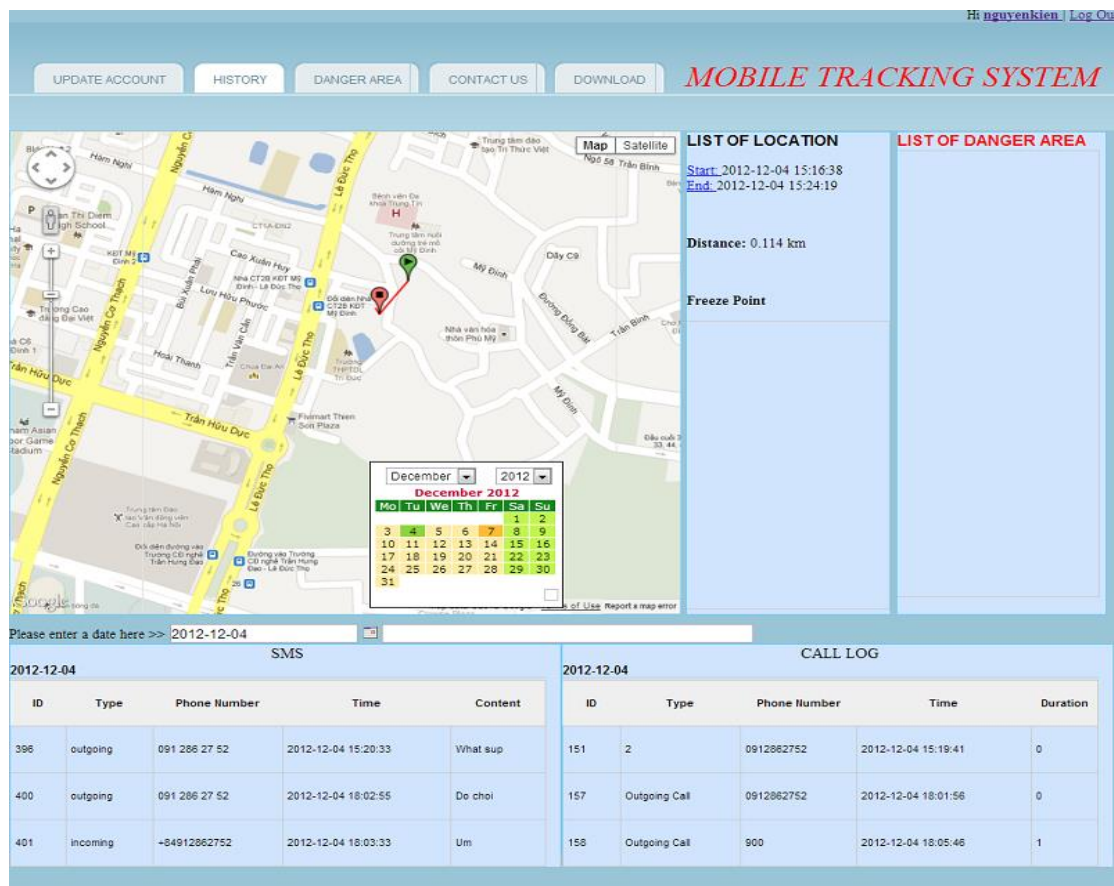


Figure 16 - Position History Screen

SMS history screen:

SMS				
2012-12-07				
ID	Type	Phone Number	Time	Content
495	incoming	+84945709689	2012-12-07 01:56:42	Hay qua
496	incoming	+84945709689	2012-12-07 01:57:42	mtjg.atg.jatjg.ajg.ajg.atpg.akgajgajpg.ajtgaajamaja5m.agj
497	incoming	18001091	2012-12-07 10:07:07	(QC) Truy cap Facebook MIEN PHI trong 30 ngay tren u phi) de tai ung dung ngay! Tu choi QC, soan TC gui 180
498	incoming	+84946273186	2012-12-07 11:01:42	Gui cho a so cua Nham nhe.
499	outgoing	+84946273186	2012-12-07 11:43:03	Call mobile 01689914561
500	outgoing	+84946273186	2012-12-07 11:43:34	Ak a goi cho thang Tung beo chua

Figure 17 - SMS History Screen

X

SMS Info

Id	495
Type	incoming
Phone Number	+84945709689
Time	2012-12-07 01:56:42
Content	<div style="border: 1px solid black; padding: 5px; min-height: 80px;"> Hay quas </div>

Figure 18 - SMS Information Detail

Call logs history screen:

CALL LOG				
2012-12-07				
ID	Type	Phone Number	Time	Duration
318	Miss Call	0945709689	2012-12-07 01:21:38	
319	Receive Call	0945709689	2012-12-07 01:21:47	3
320	Outgoing Call	0945709689	2012-12-07 01:22:06	0
336	Receive Call	18001091	2012-12-07 10:10:55	0
337	Miss Call	01237386224	2012-12-07 10:57:50	
338	Outgoing Call	+84946273186	2012-12-07 11:43:34	0
339	Miss Call	0946273186	2012-12-07 11:46:02	
340	Outgoing Call	0945709689	2012-12-07 14:36:13	3

Figure 19 - Call logs history screen

X

Call Log Info

Id	318
Type	Miss Call
Phone Number	0945709689
Time	2012-12-07 01:21:38
Duration	s

Figure 20 - Call Logs Information Detail

Detail Process:

Function Specification			
Function No	BF4A1	Function Version	1.0
Function Name	Display statistic information		
Author	MTS		
Date	10/10/2012	Priority	Normal
Description System will display all tracking information for user. Displayed information is divided by date.			
Inputs History date.			
Source Date dropdown list, choose date on list.			

Outputs
List data on chosen date, include: sms, call logs and position.
Destination
Sms & call logs: table of History screen. Position: Map on History screen.
Requires
History screen is displayed. Map is loaded.
Pre-condition
End users login by users' role.
Post-condition
<ul style="list-style-type: none"> - Sms & call logs are selected from server database and displayed on table according chosen date. - Position is selected from server database and displayed on map according chosen date.
Exception
<p>In case be failed when connect to server database to displayed data, system will:</p> <ul style="list-style-type: none"> - System is busy! Please try again later!

Table 25 - Display Statistic Information Function Specification

Activity Diagram:

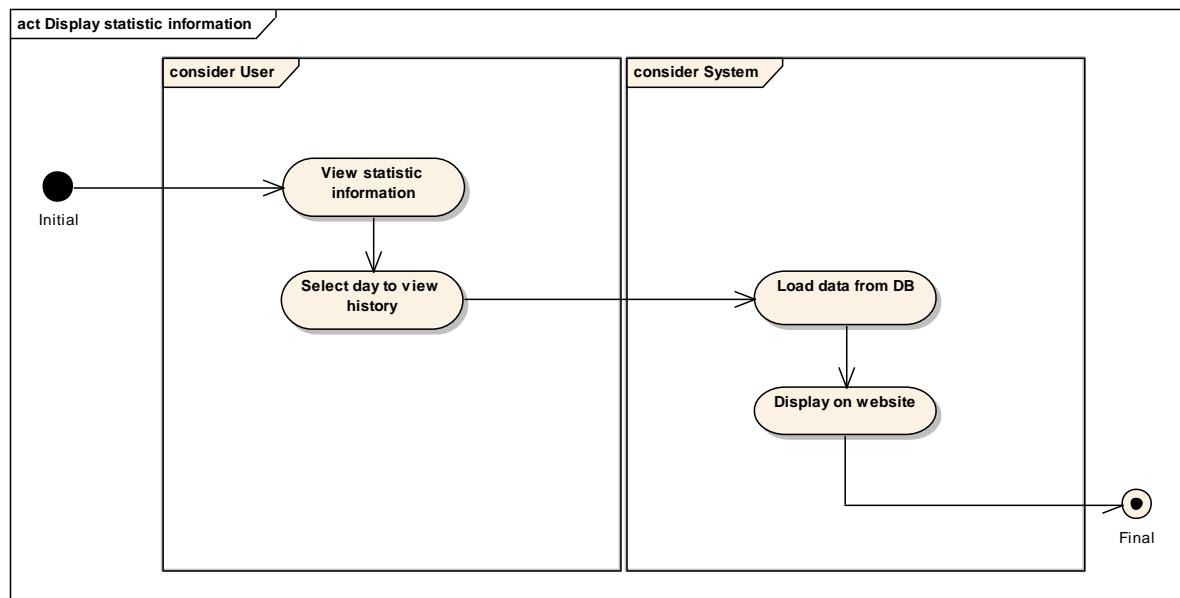


Figure 21 - Display statistic information diagram

Activity Name	Flow	Code	Description
View statistic information	BF4	A1.UP1	User select “View statistic information” link from Home screen.
Select day to view history	BF4	A1.UP2	Server store history during 30 days. But on screen, system just displays information of one day. User selects the day to view history.
Load data from DB to display	BF4	A1.SP3	System picks data of selected day from DB to display on website.

Table 26 – Display statistic information process description

3.2.7 Function “Register”

In this function, system allows user create an account to login system and use tracking service

Register Screen:

Figure 22 - Register Screen

Registration Confirm Screen

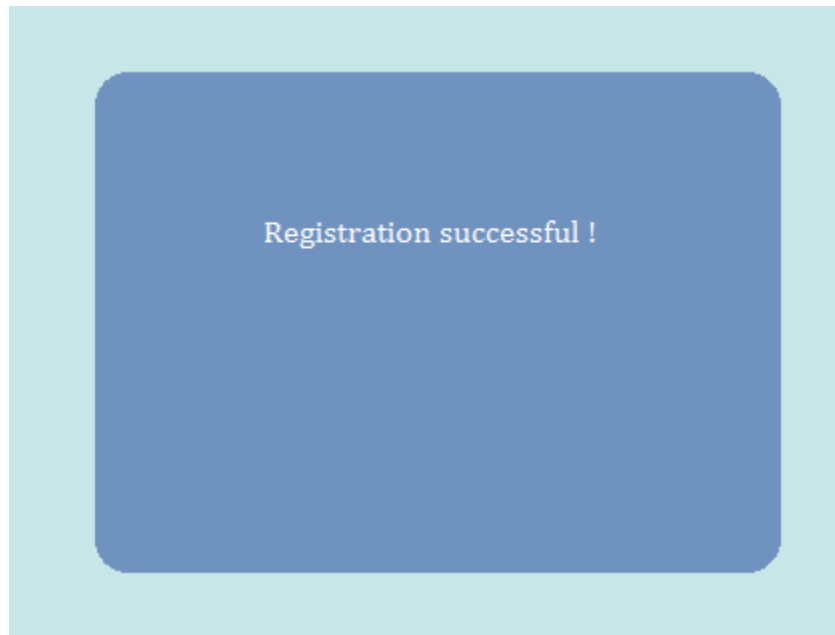


Figure 23 - Registration Confirm Screen

Detail Process:

Function Specification			
Function No	BF5A1	Function Version	1.0
Function Name	Register		
Author	MTS		
Date	10/10/2012	Priority	Normal
Description			
System allows user create an account to login system and use tracking service			
Inputs			
End users input User name, Password, Confirm password, IMEI, Phone and Email into Register form.			
Source			
User name, Password, Confirm password, IMEI, Phone and Email from text fields.			
Validated			
Item	Required	Validate	Messages
IMEI	*	15 digits	- "Tối thiểu 15 ký tự."

			- “Dữ liệu là chữ số.”
Username	*	From 6 to 12 characters. Cannot input special characters	- “Tối thiểu 6 kí tự.” - “Tối đa 12 kí tự.”
Password	*	Display(*) type. From 6 to 12 characters. Cannot input special characters	- “Tối thiểu 6 kí tự.” - “Tối đa 12 kí tự.”
Confirm password		Equal to password.	- “Dữ liệu nhập lại không hợp lệ”
Mobile	*	Only number	- “Dữ liệu là số”
Email	*	Email format.	- “Địa chỉ email không hợp lệ”

Outputs

- An account is created if validated.
- Show successful messages.

Destination

Register Screen.

Requires

Register screen is loaded.

Pre-condition

End users has not an account.

Post-condition

- An account is created on database.
- Successful message is displayed.
- User can log in by this account.

Exception

Incase be failed cause by user name or imei or email existed, show message:

- User name is existed.
- Imei is existed.
- Email is existed.

In case be failed when save to server database, system will:

- Logs exception.
- Check network to resave.

Table 27 - Register Function Specification

Activity Diagram

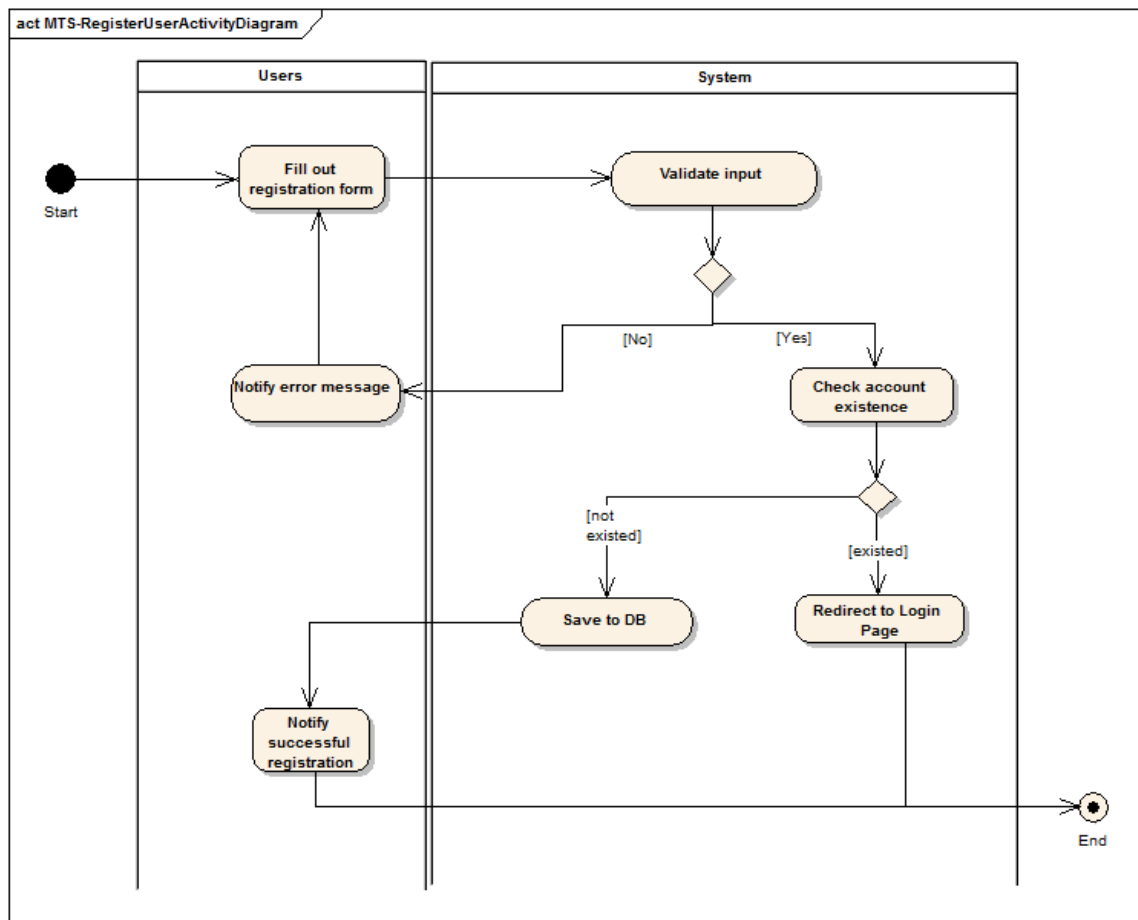


Figure 24 - Register Activity Diagram

Activity Name	Flow	Code	Description									
Fill out registration form	BF5	A1.UP 1	When using website to remote track, users have to register an account. To register, users must enter profile information to registration form.									
Validate input	BF5	A1.SP 2	After registering, information is validated follow: <table><tr><th>Item</th><th>Required</th><th>Type</th></tr><tr><td>IMEI</td><td>*</td><td>string</td></tr><tr><td>Username</td><td>*</td><td>string</td></tr></table>	Item	Required	Type	IMEI	*	string	Username	*	string
Item	Required	Type										
IMEI	*	string										
Username	*	string										

Activity Name	Flow	Code	Description		
			Password	*	string
			Mobile	*	string
			Email	*	string
Notify error messages		A1.SP 3	If validation fails system will immediately notify problems to user and remark to user re-enters information		
Check account existence		A1.SP 4	Check existence of account. In case account is existed, show error message and redirect to Login page		
Save to DB		A1.SP 5	Save all information of account to database when registered successfully.		
Redirect to Login Page	BF5	A1.SP 6	In case account already exists for the given email, system will be forwarded to the login page.		
Notify successful registration	BF5	A1.SP 7	User is registered. Success messages will be show to user to authenticate.		

Table 28 – Register Process Description

3.2.8 Function “Login”

In this function, user login to website to use system service.

Login Screen

Figure 25 - Login screen

Detail Process

Function Specification															
Function No	BF6A1	Function Version	1.0												
Function Name	Login														
Author	MTS														
Date	10/10/2012	Priority	Normal												
Description User login to website to use system service.															
Inputs <ul style="list-style-type: none">Login: End users input Username & Password into Login Form.															
Source <ul style="list-style-type: none">Login: User name and Password from text fields Login form.															
Validate <ul style="list-style-type: none">Login: <table><tr><th>Item</th><th>Required</th><th>Validate</th><th>Messages</th></tr><tr><td>Username</td><td>*</td><td>From 6 to 12 characters. Cannot input special characters</td><td><ul style="list-style-type: none">- “Tối thiểu 6 kí tự.”- “Tối đa 12 kí tự.”</td></tr><tr><td>Password</td><td>*</td><td>Display(*) type. From 6 to 12 characters. Cannot input special characters</td><td><ul style="list-style-type: none">- “Tối thiểu 6 kí tự.”- “Tối đa 12 kí tự.”</td></tr></table>				Item	Required	Validate	Messages	Username	*	From 6 to 12 characters. Cannot input special characters	<ul style="list-style-type: none">- “Tối thiểu 6 kí tự.”- “Tối đa 12 kí tự.”	Password	*	Display(*) type. From 6 to 12 characters. Cannot input special characters	<ul style="list-style-type: none">- “Tối thiểu 6 kí tự.”- “Tối đa 12 kí tự.”
Item	Required	Validate	Messages												
Username	*	From 6 to 12 characters. Cannot input special characters	<ul style="list-style-type: none">- “Tối thiểu 6 kí tự.”- “Tối đa 12 kí tự.”												
Password	*	Display(*) type. From 6 to 12 characters. Cannot input special characters	<ul style="list-style-type: none">- “Tối thiểu 6 kí tự.”- “Tối đa 12 kí tự.”												
Outputs If account is exited, user login successfully by own account.															
Destination <ul style="list-style-type: none">Login screen.															
Requires <ul style="list-style-type: none">Login screen is displayed.															
Pre-condition															

<ul style="list-style-type: none"> - Login: End users go to web by url
Post-condition <ul style="list-style-type: none"> - Login: End users login by own account and use all feature of system by users' role.
Exception <p>In case be failed when upload to server database, system will:</p> <ul style="list-style-type: none"> - Logs exception. - Check network to resend.

Table 29 - Login Function Specification

Activity Diagram

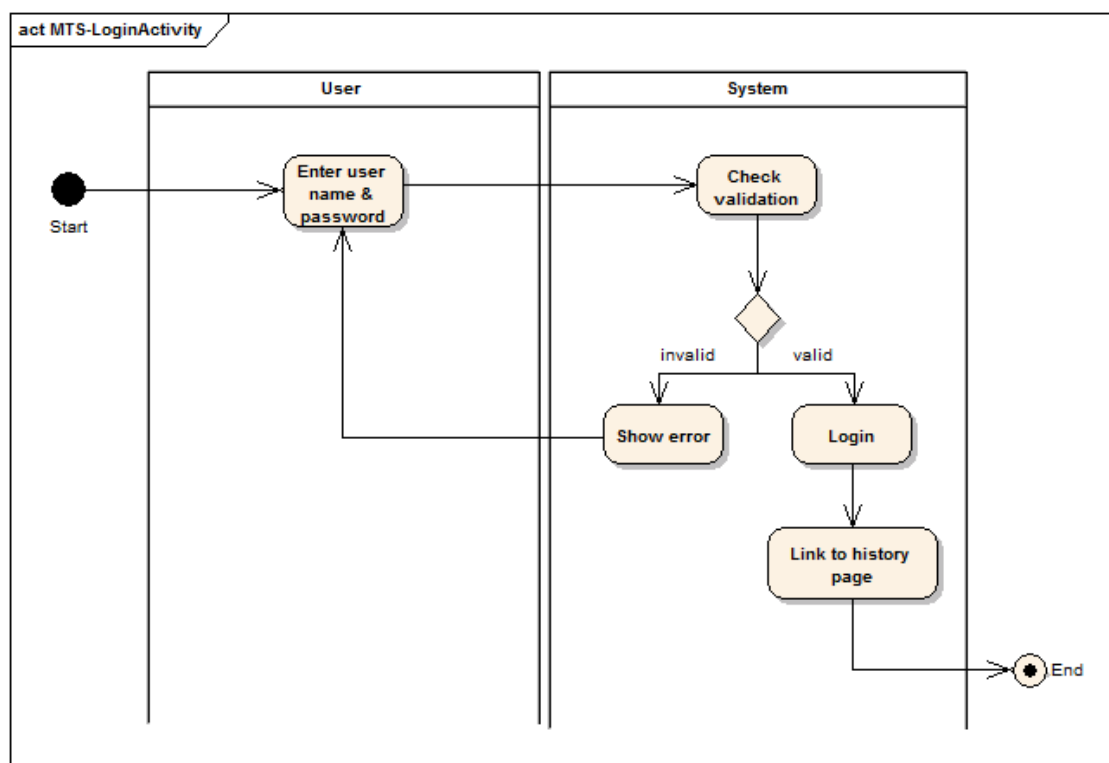


Figure 26 - Login Activity diagram

Activity Name	Flow	Code	Description
Enter user name & password	BF6	A1.UP1	When using website to remote track, users must login to system. To login, users must enter user name and password into login form.
Check validation	BF6	A1.SP2	After login, user name & password must be validated.

Activity Name	Flow	Code	Description
Show error	BF6	A1.SP3	Incase username and password are invalid, notify user by showing errors
Login	BF6	A1.SP4	In case username & password are valid, login successfully.
Link to history page	BF6	A1.SP5	After login successfully, system link to history page.

Table 30 - Login Process Description

3.2.9 Function “Update Profile”

In this function, system allows users update profile information. Updated information includes:

- User name
- IMEI
- Email
- Mobile

Update Profile Screen

Figure 27 - Update Profile Screen

Detail Process

Function Specification																							
Function No	BF6A2	Function Version	1.0																				
Function Name	Update Profile																						
Author	MTS																						
Date	10/10/2012	Priority	Normal																				
Description User can update profile in case having changes.																							
Inputs <ul style="list-style-type: none">- Profile Update: When login successfully, input to update information about account: User name, IMEI, Mobile, Email of Profile Form.																							
Source <ul style="list-style-type: none">- Profile Update: User name, IMEI, Mobile, Email from text field of Profile form.																							
Validate <ul style="list-style-type: none">- Profile Update: <table><tr><th>Item</th><th>Required</th><th>Validate</th><th>Messages</th></tr><tr><td>IMEI</td><td>*</td><td>15 digits.</td><td><ul style="list-style-type: none">- “Tối thiểu 15 kí tự.”- “Dữ liệu là chữ số”</td></tr><tr><td>Mobile</td><td>*</td><td>Only number</td><td><ul style="list-style-type: none">- “Dữ liệu là số”</td></tr><tr><td>Email</td><td>*</td><td>Email form</td><td><ul style="list-style-type: none">- “Địa chỉ email không hợp lệ”</td></tr><tr><td>User name</td><td>*</td><td>From 6 to 12 characters. Cannot input special characters</td><td><ul style="list-style-type: none">- “Tối thiểu 6 kí tự”- “Tối đa 12 kí tự”</td></tr></table>				Item	Required	Validate	Messages	IMEI	*	15 digits.	<ul style="list-style-type: none">- “Tối thiểu 15 kí tự.”- “Dữ liệu là chữ số”	Mobile	*	Only number	<ul style="list-style-type: none">- “Dữ liệu là số”	Email	*	Email form	<ul style="list-style-type: none">- “Địa chỉ email không hợp lệ”	User name	*	From 6 to 12 characters. Cannot input special characters	<ul style="list-style-type: none">- “Tối thiểu 6 kí tự”- “Tối đa 12 kí tự”
Item	Required	Validate	Messages																				
IMEI	*	15 digits.	<ul style="list-style-type: none">- “Tối thiểu 15 kí tự.”- “Dữ liệu là chữ số”																				
Mobile	*	Only number	<ul style="list-style-type: none">- “Dữ liệu là số”																				
Email	*	Email form	<ul style="list-style-type: none">- “Địa chỉ email không hợp lệ”																				
User name	*	From 6 to 12 characters. Cannot input special characters	<ul style="list-style-type: none">- “Tối thiểu 6 kí tự”- “Tối đa 12 kí tự”																				
Outputs View profile to update.																							
Destination <ul style="list-style-type: none">- Profile screen																							
Requires																							

<ul style="list-style-type: none"> - Profile screen is displayed with data if login successfully.
Pre-condition <ul style="list-style-type: none"> - Profile Update: End users login by users' role.
Post-condition <ul style="list-style-type: none"> - Profile Updated: Profile is updated to server database
Exception <p>In case be failed when upload to server database, system will:</p> <ul style="list-style-type: none"> - Logs exception. - Check network to resend.

Table 31 - Update Profile Function Specification

Activities Diagram

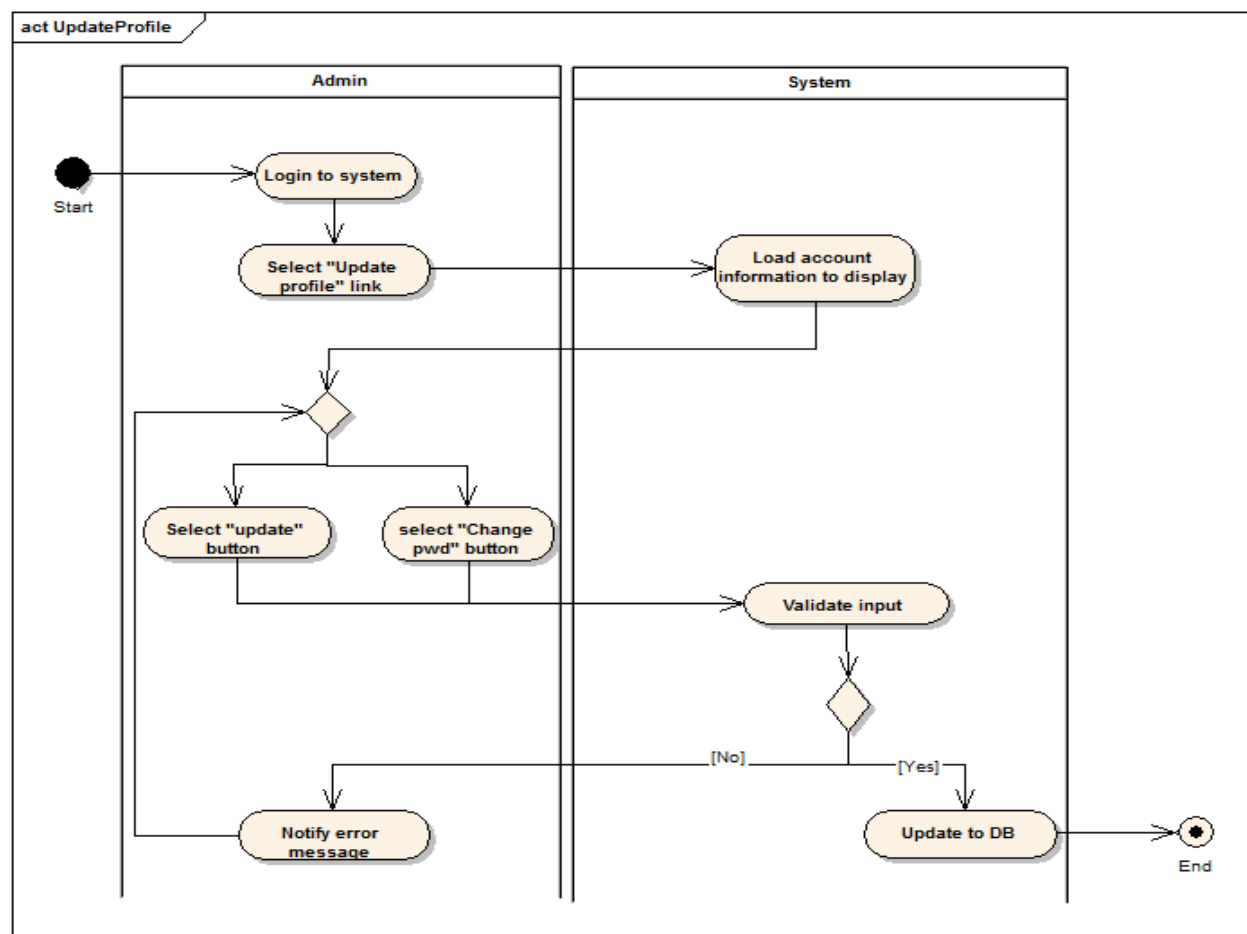


Figure 28 - Update Profile Activity Diagram

Activity Name	Flow	Code	Description
Login to system	BF6	A2.UP1	Users must have login to system by registered account.
Select “Update Profile” link	BF7	A1.SP2	After login successfully, Select “Update Profile ” link on menu of home page.
Load account information to display	BF7	A1.UP3	Profile information are loaded when user click on “Update Profile” on home page menu. Profile information includes: User name, IMEI, Mobile, Email.
Select “Update” button	BF7	A1.SP4	Select “Update” to update changed account information.
Select “Change pwd” button	BF7	A1.UP5	Select “Change pwd” to change current password.
Validate input	BF7	A1.SP6	When input changed data, system will validate information.
Notify error message	BF7	A1.UP7	Incase data inputs are invalid, notify user by showing errors
Update to database	BF7	A1.UP8	Update all information of account to database after click one of button: Update, Change pwd.

Table 32 - Update Profile Activity Description

3.2.10 Function “Manage Account”

In this function, system allows admin manage account information. Account managed information includes:

- User name
- Role
- IMEI
- Mobile

- Email

Manage Account Screen:

Hi [admin1234](#) | [Log Out](#)

MOBILE TRACKING SYSTEM

ID	User Name	Password	Role	Imei	Phone Number	Email
7	giangvh001	e10adc3949ba59abbe56e057f20f883e	2	1111111111111111	090111111111	giangvh001@yahoo.com.vn
11	admin1234	e10adc3949ba59abbe56e057f20f883e	1	9999999999999999	090999999999	admin1234@yahoo.com.vn
4	nguyenkien	e10adc3949ba59abbe56e057f20f883e	2	358490041283939	0912862752	white1990black@yahoo.com
5	kyenonline	e10adc3949ba59abbe56e057f20f883e	2	0000000000000000	0912862751	white1980black@yahoo.com
8	giangvh002	a5bd4b4c9fb4308176ccb8e72a2b674b	2	2222222222222222	090222222222	giangvh002@yahoo.com.vn
10	giangvh003	9fdc385474dfef58e099b90421a87e7d	2	3333333333333333	090333333333	giangvh003@yahoo.com.vn

Figure 29 - Update account screen

Account Info

Id	7
User Name	giangvh001
Role	User ▼
IMEI	1111111111111111
Mobile	090111111111
Email	giangvh001@yahoo.com.vn

Figure 30 - Account Information Detail

Detail Process:

Function Specification			
Function No	BF7A1	Function Version	1.0

Function Name	Manage Account		
Author	MTS		
Date	10/10/2012	Priority	Normal
Description			
System allows admin update account information.			
Inputs			
Admin input User name, Role, IMEI, Mobile, Email into Account Management popup form.			
Source			
Password and Role from text fields of Account Management popup form.			
Outputs			
Account information are changed.			
Destination			
Account Management screen.			
Requires			
Account Management screen is displayed. Account information is loaded.			
Pre-condition			
Adminlogin by admin's role.			
Post-condition			
Account information are updated and stored in server database.			
Exception			
In case be failed when upload to server database, system will:			
<ul style="list-style-type: none"> - Logs exception. - Check network to resave. 			

Table 33 - Update Account Function Specification

Activity Diagram

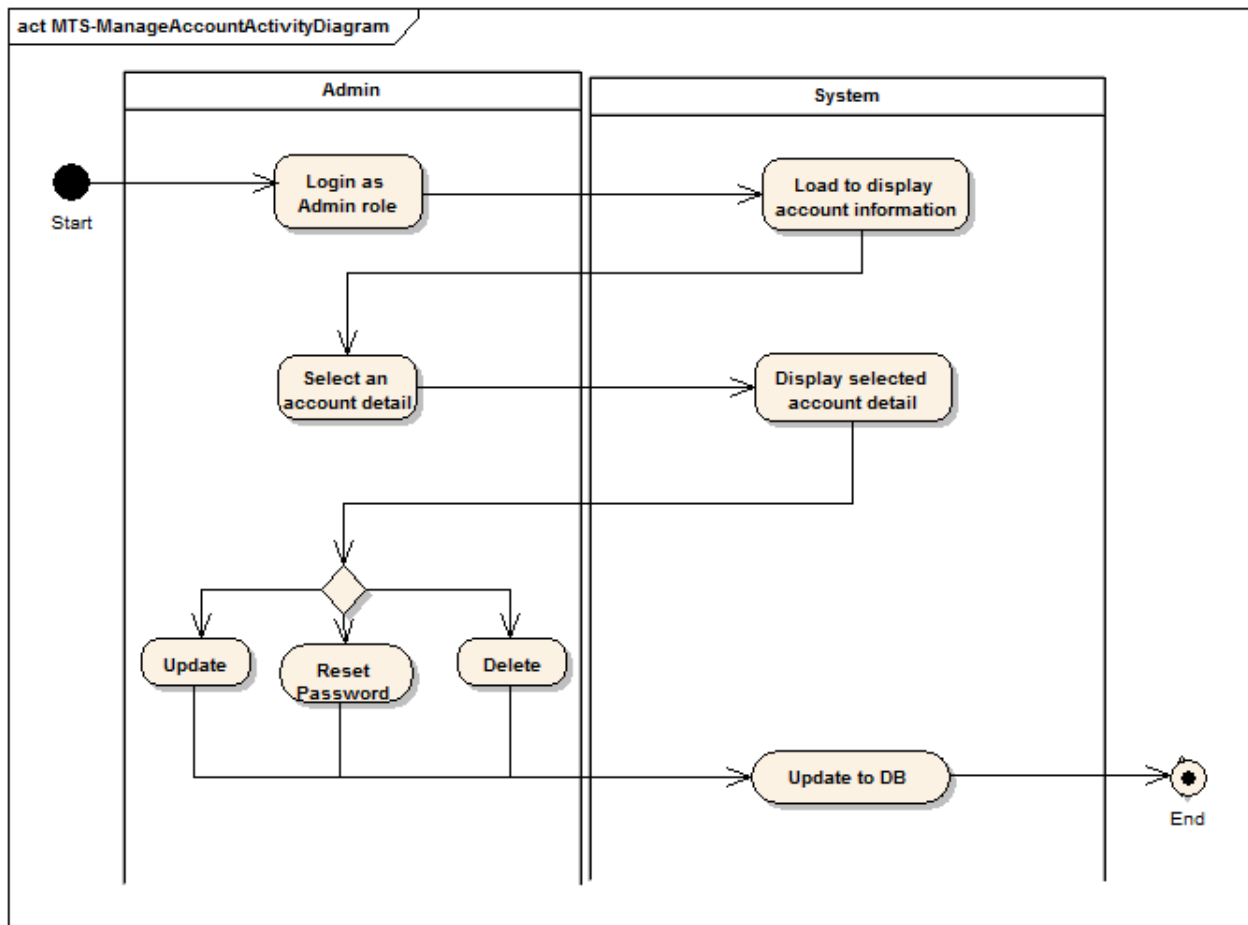


Figure 31 - Update Account Activity Diagram

Activity Name	Flow	Code	Description
Login as admin role	BF7	A1.UP1	To manage accounts, admin must have login to system as administrator.
Load to display account information	BF7	A1.SP2	System connect database to select all account. Then display list of accounts to screen. Paging when list view is too long.
Select an account detail	BF7	A1.UP3	When view any account detail, admin can click on username of account link on list to select.
Display selected account detail	BF7	A1.SP4	Information detail of selected account is displayed immediately after click on. Display popup.

Activity Name	Flow	Code	Description
Update	BF7	A1.UP5	Admin can update account that have just selected. Edit to change some information of user.
Reset password	BF7	A1.SP6	Admin can reset password of account that have just selected.
Delete	BF7	A1.UP7	Admin can delete account that have just selected. Click "Delete" to remove selected account from system
Update to database	BF7	A1.UP8	Update all information of account to database after click one of button: Update, Delete, Reset password.

Table 34 – Manage Account Process Description

3.3 Non – Functional Requirement

3.3.1 Usability

- The website interface is clear, simple and out-standing.
- The website is designed with friendly user interface. Input will be validated. The user should only have general computer use knowledge. Error messages will be displayed in case of an invalid request. Users need no time for training or reading guideline.
- Searching tool should be easy to use.
- The website must provide a help page to support novice users.
- Users can use main functionality of the system without logging in to the system.

3.3.2 Reliability

- The database must be backed up regularly and can be recovered if necessary
- Mean time between failures (MTBF): The mean time between each failure is expected to be about 480 hours (20 days) of using.
- Mean time to repair (MTTR): Immediately when admin finds out problem or website is attacked by someone. Average 1 day.
- The maximum bugs per function are 2 bugs/a function. The average number of bugs in the software is about to be around 0.5 bugs.

Bug or Defect Rate

Bug	Definition
-----	------------

Bug	Definition
Minor	Bugs related to GUI
Significant	Bugs related to minor business logic
Critical	Function can't execute correctly, completely loss of data, disconnect to the server

Table 35 – Bugs or Defect Rate

3.3.3 Availability

- The system will be available 24/24 and should be closed 1 day for maintaining every two months
- Percentage of time available is about 95-97%. Because the system uses client/server protocol through Internet network, we cannot control cases of force majeure such as power failure, disconnected network...

3.3.4 Security

- Privacy: the system should provide protection method for protecting user information from outside or from other users. All the information of users must not be available for anyone or software that is not part of the system. User password is also invisible for the system administrator.
- The system must provide methods to prevent common security attacks. E.g. SQL injection...
- Application is installed underground.
- Transformation of confidential information must be encrypted.
- The system must provide secure methods for users to recovery their password, including the interference of system's admin.

3.3.5 Maintainability

❖ Coding standards and naming conventions

- Output of the project must include coding standards and naming conventions documentations. Implementation code must be easy to maintain.
- All code must be clearly commented, including class, method documentations.
- If some components are reused, the documentations of those components must also be included.

❖ Design

- The design of the system must be loosely coupled that chances on some module will not affect others.

❖ **Logging**

- All the errors should be logged, supporting for bug fixing and maintenance.
- All strange or sensitive situations should also be logged.

❖ **Email**

- The system must provide an email address to receive customer feedback or send emails.

3.3.6 Portability

- Website and smart phone can be used by any people with a browser and an Internet connection.

3.3.7 Performance

- Response time under 10 seconds

IV. Chapter IV - Software Architecture Design

4.1 System Architecture Design

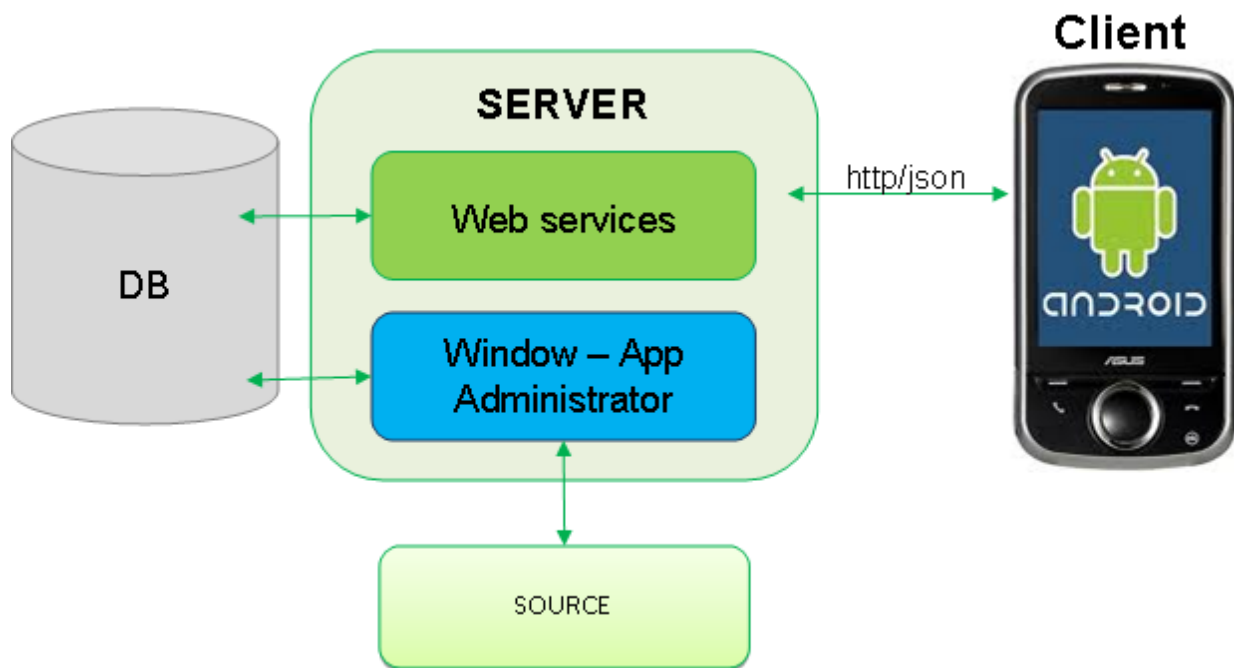


Figure 32 - System Overview

4.1.1 Mobile

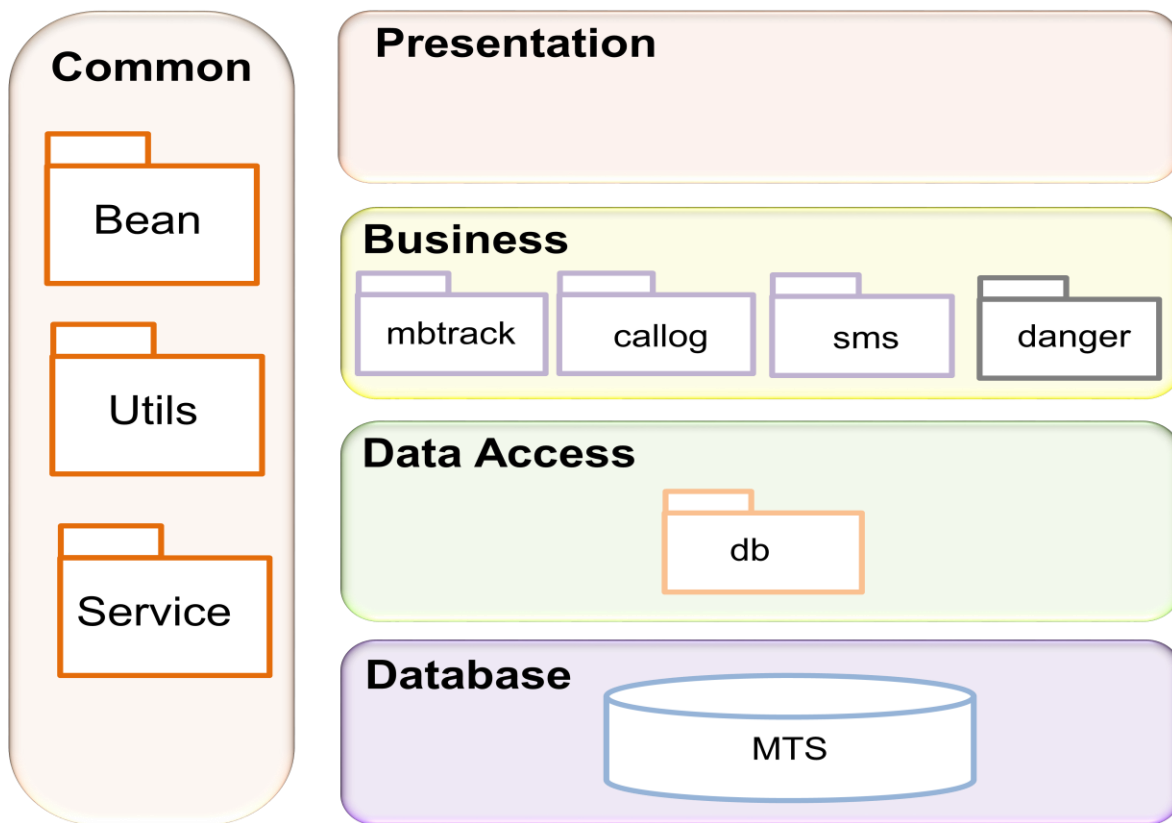


Figure 33 - Mobile Architecture Design

ID	Package Name	Description	Notes
01	Common Bean	Contains Bean Classes	
02	Common Utils	Contains Utility Classes	
03	Common Service	Contains Service Classes	
04	Business mbtrack		
05	Business sms	Contains SMSTracking Classes	
06	Business callog	Contains Call Logs Tracking Classes	
07	Business danger		
08	Data Access DB	Contains Database Interaction Classes	

Table 36 - Mobile Packages

4.1.2 Server

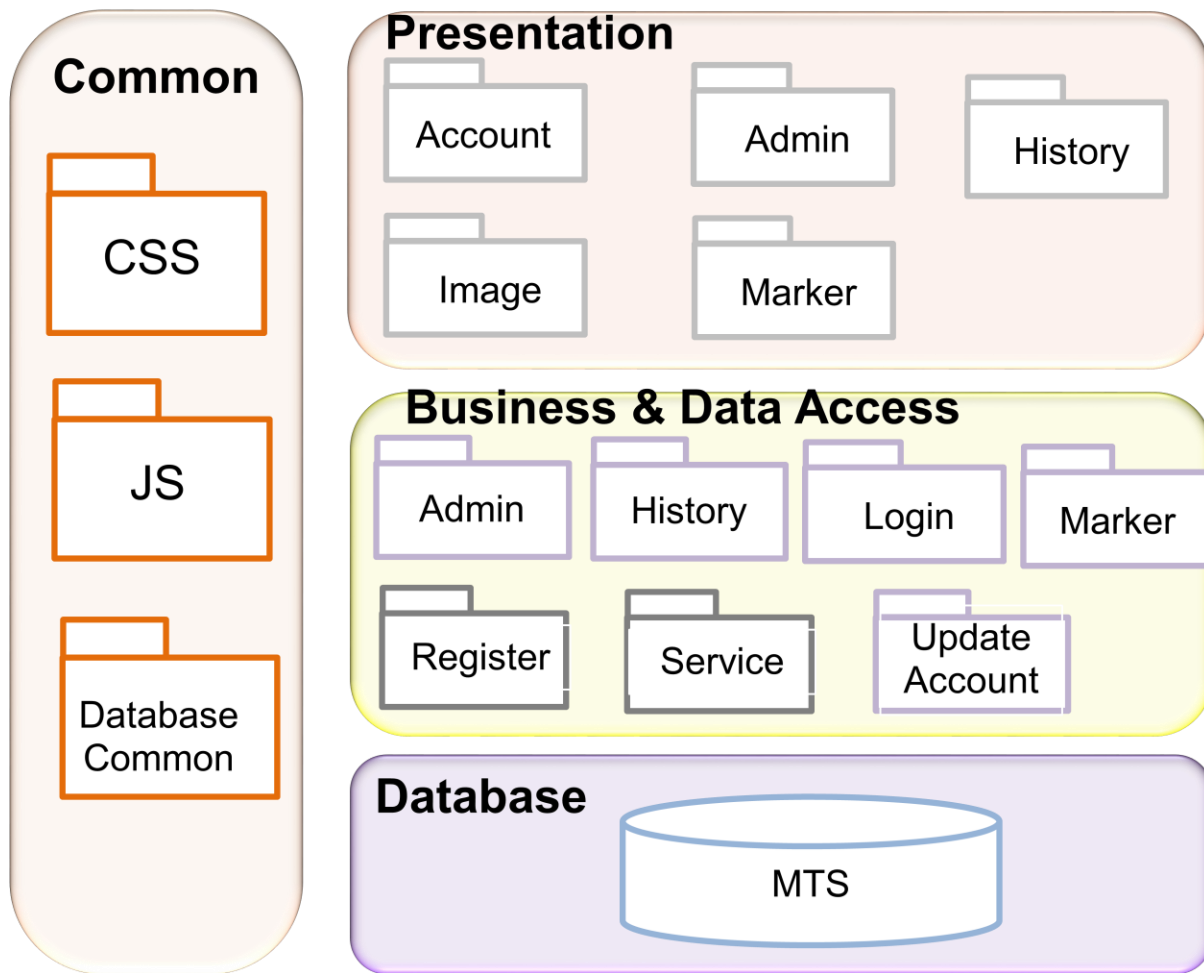


Figure 34 - Server Architecture Design

ID	Package Name	Description	Notes
01	Common CSS	Contains Style Sheet Files	
02	Common JS	Contains JavaScript Files	
03	Common Database Common	Contains Database Access Classes	
04	Presentation History	Contains Interface Files of History Process	
05	Presentation Image	Contains Image Files	
06	Presentation Marker	Contains Interface Files of Marker Process	
07	Presentation Account	Contains Interface Files of Account Management Process	
08	Presentation Admin	Contains Interface Files of Admin Process	
09	Business History	Contains All Files of History Process Logic	
10	Business Login	Contains All Files of Login Process Logic	
11	Business Marker	Contains All Files of Marker Process Logic	
12	Business Register	Contains All Files of Register Process Logic	

13	Business Service	Contains Service Files	
14	Business Admin	Contains All Files of Admin Process Logic	
15	Business Update Account	Contains All Files of Update Account Process Logic	

Table 37 - Server Packages

4.2 Class Diagram

4.2.1 Mobile

4.2.1.1 SMS

4.2.1.1.1 Class diagram

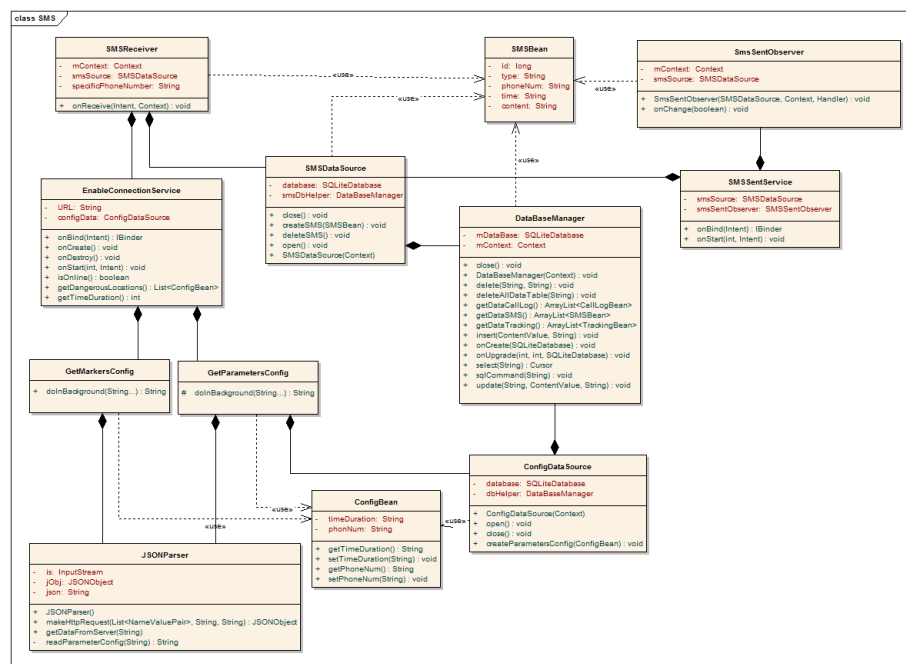


Figure 35 - SMS Class Diagram

4.2.1.1.2 Class explanation

4.2.1.1.2.1 SMSReceiver

Package	com.android.mts.sms														
Class	SMSReceiver														
Attribute	<table border="1"> <thead> <tr> <th>No</th><th>Name</th><th>Type</th><th>Description</th></tr> </thead> <tbody> <tr> <td>01</td><td>mContext</td><td>Context</td><td></td></tr> <tr> <td>02</td><td>smsSource</td><td>SMSDataSource</td><td></td></tr> </tbody> </table>			No	Name	Type	Description	01	mContext	Context		02	smsSource	SMSDataSource	
No	Name	Type	Description												
01	mContext	Context													
02	smsSource	SMSDataSource													

	03	specificPhoneNumber	String											
Method	<table> <tr> <th>No</th><th>Name</th><th>Return Type</th><th>Parameter</th><th>Description</th></tr> <tr> <td>01</td><td>onReceive(Context context, Intent intent)</td><td></td><td>context, intent</td><td></td></tr> </table>				No	Name	Return Type	Parameter	Description	01	onReceive(Context context, Intent intent)		context, intent	
No	Name	Return Type	Parameter	Description										
01	onReceive(Context context, Intent intent)		context, intent											

Table 38 - SMSReceiver

4.2.1.1.2.2 SMSDataSource

Package	com.android.mts.sms				
Class	SMSDataSource				
Attribute					
	No	Name	Type	Description	
	01	database	SQLiteDatabase		
	02	smsDbHelper	DataBaseManager		
Method					
	No	Name	Return Type	Parameter	Description
	01	close()			
	02	createSMS(SMSBean sms)		sms	
	03	deleteSMS()			
	04	open()			
	05	SMSDataSource(Context context)		context	

Table 39 - SMSDataSource

4.2.1.1.2.3 SMSSentObserver

Package	com.android.mts.sms			
Class	SMSSentObserver			

Attribute					
	No	Name	Type	Description	
	01	mContext	Context		
	02	smsSource	SMSDataSource		
Method					
	No	Name	Return Type	Parameter	Description
	01	onChange(boolean selfChange)		selfChange	
	02	SmsSentObserver(Handler handler, Context ctx, SMSDataSource smsSource)		Handler,ctx ,smsSource	

Table 40 - SMSentObserver

4.2.1.1.2.4 EnableConnectionService

Package	com.android.mts.service				
Class	EnableConnectionService				
Attribute					
	No	Name	Type	Description	
	01	URL	String		
	02	configData	ConfigDataSource		
Method					
	No	Name	Return Type	Parameter	Description
	01	isOnline()	T/F		
	02	onBind()			
	03	onCreate()			
	04	onDestroy()			
	05	onStart(Intent intent, int startid)		Intent, startid	
	06	getDangerousLocation()	List		
	07	getTimeDuaration	int		

Table 41 – EnableConnectionService

4.2.1.1.2.5 DataBaseManager

Package	com.android.mts.db				
Class	DataBaseManager				
Attribute					
	No	Name	Type	Description	
	01	mContext	Context		
	02	mDataBase	SQLiteDatabase		
Method					
	No	Name	Return Type	Parameter	Description
	01	DataBaseManager(Context context)		context	Constructor takes and keeps a reference of the passed context in order to access to the application assets and resources.
	02	select(String query)	Cursor	query	Return cursor with the results
	03	getDataTracking()	Array		
	04	getDataSMS()	Array		Get data from table sms
	05	getDataCallLog()	Array		Get data from table call log
	06	deleteAllDataTable(String tableName)		tableName	
	07	insert(String table, ContentValues values)		table, values	
	08	delete(String table, String where)		table, where	
	09	update(String table, ContentValues values, String where)		table, values, where	
	10	sqlCommand(String command)		command	Make a raw query

11	close()			
12	onCreate(SQLiteDatabase db)		db	
13	onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)		db, oldVersion, newVersion	

Table 42 - DatabaseManager

4.2.1.1.2.6 SMSSentService

Package	com.android.mts.sms				
Class	SMSSentService				
Attribute					
	No	Name	Type	Description	
	01	smsSentObserver	SMSSentObserver		
	02	smsSource	SMSDataSource		
Method					
	No	Name	Return Type	Parameter	Description
	01	onStart(Intent intent, int startId)		intent, startId	
	02	onBind(Intent intent)		intent	

Table 43 - SMSSentService

4.2.1.1.2.7 JSONParser

Package	com.android.mts.utils																			
Class	JSONParser																			
Attribute	<table><tr><th>No</th><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>01</td><td>is</td><td>InputStream</td><td></td></tr><tr><td>02</td><td>jObj</td><td>JSONObject</td><td></td></tr><tr><td>03</td><td>Json</td><td>String</td><td></td></tr></table>				No	Name	Type	Description	01	is	InputStream		02	jObj	JSONObject		03	Json	String	
No	Name	Type	Description																	
01	is	InputStream																		
02	jObj	JSONObject																		
03	Json	String																		

Method					
	No	Name	Return Type	Parameter	Description
	01	JSONParser()			Contructor
	02	makeHttpRequest(String url, String method, List<NameValuePair> params)	JSONObject		Get json from url
	03	getDataFromServer(String url)	JSONArray	url	Get configuration data from server
	04	readParameterConfig(String url)	String	url	

Table 44 - JSONParser

4.2.1.1.2.8 ConfigDataSource

Package	com.android.mts.config				
Class	ConfigDataSource				
Attribute					
	No	Name	Type	Description	
	01	Database	SQLiteDatabase		
	02	dbHelper	DatabaseManager		
Method					
	No	Name	Return Type	Parameter	Description
	01	close()			
	02	ConfigDataSource(Context context)		context	
	03	createParameterConfig(ConfigBean config)		config	
	04	open()			

Table 45 - ConfigDataSource

4.2.1.1.2.9 GetParametersConfig

Package	com.android.mts.service
---------	-------------------------

Class	GetParametersConfig				
Attribute					
Method					
	01	doInBackground(String...params)	String	params	

Table 46 - GetParamatersConfig

4.2.1.1.2.10 ConfigBean

Package	com.android.mts.bean																												
Class	ConfigBean																												
Attribute	<table><tr><th>No</th><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>01</td><td>phoneNum</td><td>String</td><td></td></tr><tr><td>02</td><td>timeDuration</td><td>String</td><td></td></tr></table>				No	Name	Type	Description	01	phoneNum	String		02	timeDuration	String														
No	Name	Type	Description																										
01	phoneNum	String																											
02	timeDuration	String																											
Method	<table><tr><th>No</th><th>Name</th><th>Return Type</th><th>Parameter</th><th>Description</th></tr><tr><td>01</td><td>getPhoneNum()</td><td>String</td><td></td><td></td></tr><tr><td>02</td><td>getTimeDuration()</td><td>String</td><td></td><td></td></tr><tr><td>03</td><td>setPhoneNum(String phoneNum)</td><td></td><td>phoneNum</td><td></td></tr><tr><td>04</td><td>setTimeDuation(String timeDuration)</td><td></td><td>timeDuration</td><td></td></tr></table>				No	Name	Return Type	Parameter	Description	01	getPhoneNum()	String			02	getTimeDuration()	String			03	setPhoneNum(String phoneNum)		phoneNum		04	setTimeDuation(String timeDuration)		timeDuration	
No	Name	Return Type	Parameter	Description																									
01	getPhoneNum()	String																											
02	getTimeDuration()	String																											
03	setPhoneNum(String phoneNum)		phoneNum																										
04	setTimeDuation(String timeDuration)		timeDuration																										

Table 47 - ConfigBean

4.2.1.1.2.11 SMSBean

Package	com.android.mts.bean
Class	SMSBean

Attribute				
	No	Name	Type	Description
	01	id	Long	Id of sms
	02	type	String	Sms type
	03	phoneNum	String	Sent/received sms phone number
	04	time	String	Sent /Received sms time
	05	content	String	Content of sms
Method				

Table 48– SMSBean

4.2.1.1.2.12GetMarkerConfig

Package	com.android.mts.service				
Class	GetMarkerConfig				
Attribute					
Method	No	Name	Return Type	Parameter	Description
	01	doInBackground(String...params)	String	params	

Table 49 – GetMarkerConfig

4.2.1.2 Mobile Track

4.2.1.2.1 Class diagram

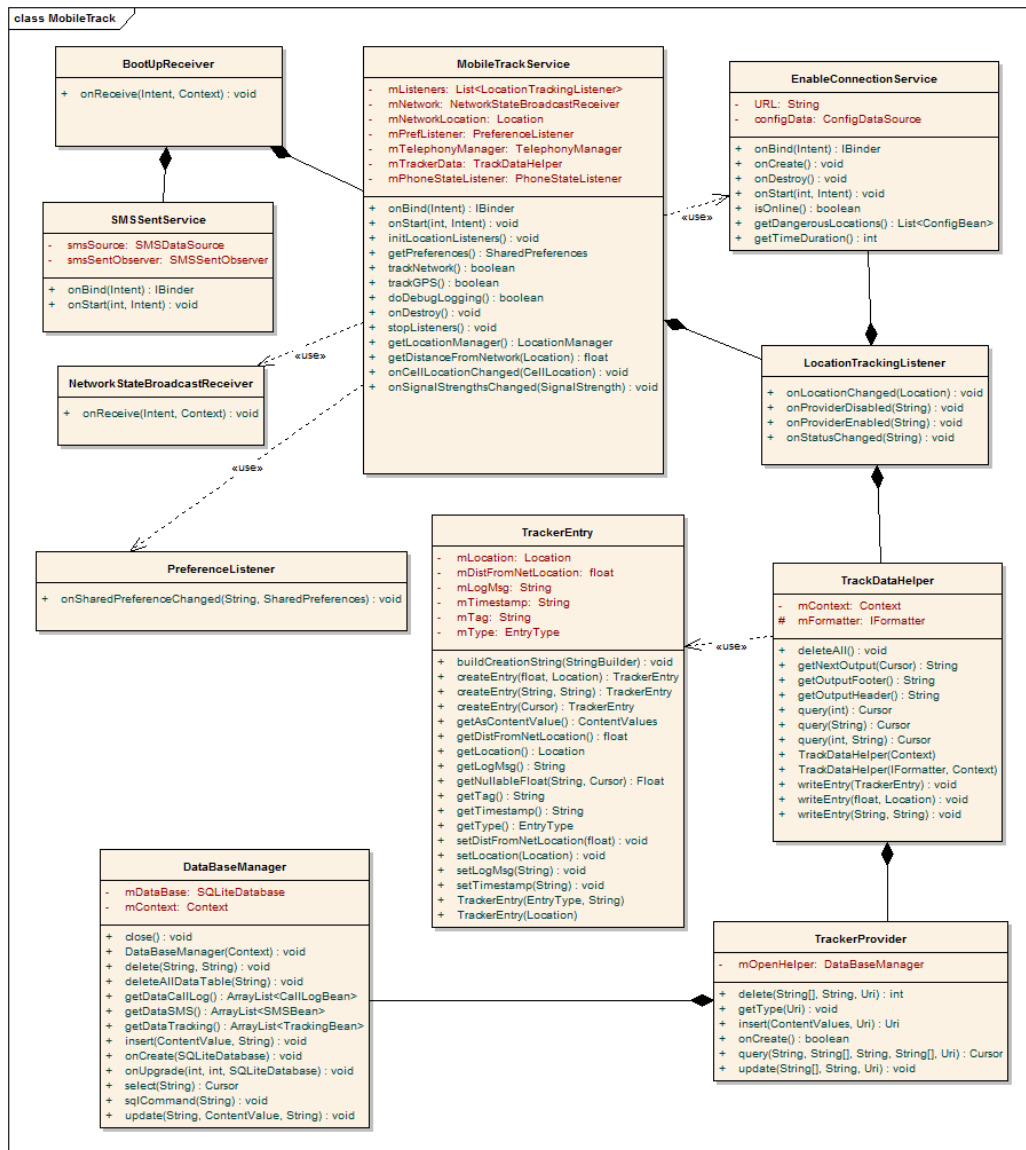


Figure 36 - Mobile Track Class Diagram

4.2.1.2.2 Class explanation

4.2.1.2.2.1 NetworkStateBroadcastReceiver

Package	com.android.mts.mbtrack
Class	NetworkStateBroadcastReceiver
Attribute	

Method					
	No	Name	Return Type	Parameter	Description
	01	onReceive(Context context, Intent intent)		Context, intent	Listener + recorder for mobile or wifi updates

Table 50 - NetworkStateBroadcastReceiver

4.2.1.2.2.2 BootUpReceiver

Package	com.android.mts.service				
Class	BootUpReceiver				
Attribute					
Method					
	No	Name	Return Type	Parameter	Description
	01	onReceive(Context context, Intent intent)		Context, intent	

Table 51 - BootUpReceiver

4.2.1.2.2.3 PreferenceListener

Package	com.android.mts.mbtrack				
Class	PreferenceListener				
Attribute					
Method					
	No	Name	Return Type	Parameter	Description
	01	onSharedPreferencesChanged(SharedPreferences sharedPreferences, String key)		shared Preferences, key	

Table 52 - PreferenceListener

4.2.1.2.2.4 SMSSentService

Package	com.android.mts.sms				
Class	SMSSentService				
Attribute					
	No	Name	Type	Description	
	01	smsSentObserver	SMSSentObserver		
	02	smsSource	SMSDataSource		
Method					
	No	Name	Return Type	Parameter	Description
	01	onStart(Intent intent, int startId)		intent, startId	
	02	onBind(Intent intent)		intent	

Table 53 - SMSSentService

4.2.1.2.2.5 MobileTrackService

Package	com.android.mts.sms																																		
Class	SMSSentService																																		
Attribute	<table> <tr> <th>No</th><th>Name</th><th>Type</th><th>Description</th></tr> <tr> <td>01</td><td>mListeners</td><td>List</td><td></td></tr> <tr> <td>02</td><td>mNetwork</td><td>NetworkState BroadcastReceiver</td><td>Handlers and Receivers for phone and network state</td></tr> <tr> <td>03</td><td>mNetworkLocation</td><td>Location</td><td></td></tr> <tr> <td>04</td><td>mPhoneStateListener</td><td>PhoneStateListener</td><td></td></tr> <tr> <td>05</td><td>mPrefListener</td><td>PreferenceListener</td><td></td></tr> <tr> <td>06</td><td>mTelephonyManager</td><td>TelephonyManager</td><td></td></tr> <tr> <td>07</td><td>mTrackerData</td><td>TrackDataHelper</td><td></td></tr> </table>			No	Name	Type	Description	01	mListeners	List		02	mNetwork	NetworkState BroadcastReceiver	Handlers and Receivers for phone and network state	03	mNetworkLocation	Location		04	mPhoneStateListener	PhoneStateListener		05	mPrefListener	PreferenceListener		06	mTelephonyManager	TelephonyManager		07	mTrackerData	TrackDataHelper	
No	Name	Type	Description																																
01	mListeners	List																																	
02	mNetwork	NetworkState BroadcastReceiver	Handlers and Receivers for phone and network state																																
03	mNetworkLocation	Location																																	
04	mPhoneStateListener	PhoneStateListener																																	
05	mPrefListener	PreferenceListener																																	
06	mTelephonyManager	TelephonyManager																																	
07	mTrackerData	TrackDataHelper																																	

Method					
	No	Name	Return Type	Parameter	Description
	01	doDebugLogging()	T/F		
	02	getDistanceFromNetwork (Location location)	float	location	Determine the current distance from given location
	03	getLocationManager	Location Manager		
	04	getPreferences()	Shared Preferences		
	05	initLocationListeners()			
	06	onBind(Intent intent)	IBinder		
	07	onCellLocationChanged (CellLocation location)		location	
	08	onDestroy()			Shut down this service
	09	onSignalStrengthsChanged (SignalStrength signalStrength)		signalStrength	
	10	onStart(Intent intent, int startId)		Intent, startId	
	11	stopListeners()			De-registers all location listeners, closes persistent storage
	12	trackGPS()	T/F		
	13	trackNetwork()	T/F		

Table 54 - MobileTrackService

4.2.1.2.2.6 LocationTrackingListener

Package	com.android.mts.mbtrack
Class	LocationTrackingListener
Attribute	

Method					
	No	Name	Return Type	Parameter	Description
	01	onLocationChanged(Location location)		location	Detail of location update to tracking file
	02	onProviderDisabled(String provider)		provider	Update to tracking file
	03	onProviderEnable(String provider)		provider	Update to tracking file
	04	onStatusChanged(String provider, int status, Bundle extras)		Provider,status, extras	Update to tracking file

Table 55 - LocationTrackingListener

4.2.1.2.2.7 TrackerProvider

Package	com.android.mts.mbtrack				
Class	TrackerProvider				
Attribute					
	No	Name	Type	Description	
	01	mOpenHelper	DataBaseManager		
Method					
	No	Name	Return Type	Parameter	Description
	01	delete(Uri uri, String selection, String[] selectionArgs)	Int	uri, selection, selectionArgs	
	02	onCreate()	T/F		
	03	getType(Uri uri)	String	uri	
	04	insert(Uri uri, ContentValues values)	Uri	uri, values	
	05	query(Uri uri,String[] projection,String selection, String[] selectionArgs, String sortOrder)	Cursor	uri, projection, selection, selectionArgs, sortOrder	
	06	update(Uri uri, ContentValues values, String selection, String[] selectionArgs)	Int	uri, values, selection, selectionArgs	

	selectionArgs)			
--	----------------	--	--	--

Table 56 - TrackerProvider

4.2.1.2.2.8 TrackerEntry

Package	com.android.mts.mbtrack				
Class	TrackerEntry				
Attribute					
	No	Name	Type	Description	
	01	mDistFromNetLocation	Float		
	02	mLocation	Location		
	03	mLogMsg	String		
	04	mTag	String		
	05	mTimestamp	String		
	06	mType	EntryType		
Method					
	No	Name	Return Type	Parameter	Description
	01	buildCreationString (StringBuilder builder)		builder	
	02	createEntry (Cursor cursor)	Tracker Entry	cursor	
	03	createEntry (String tag, String msg)	Tracker Entry	Tag,msg	Create a rackerEntry from a log msg
	04	createEntry (Location loc, float distFromNetLocation)	Tracker Entry	Loc,distFrom NetLocation	Create a rackerEntry from a Location
	05	getAsContentValues()	Content Values		
	06	getDistFromNetLocation()	Float		
	07	getLocation()	Location		
	08	getLogMsg()	String		
	09	getTag()	String		
	10	getTimestamp()	String		
	11	getType()	Entry Type		

12	setLogMsg(String msg)		msg	
13	setDistFromNetLocation(float distFromNetLocation)		distFromNetLocation	
14	setTimestamp(String timestamp)		timestamp	
15	setLocation(Location location)		location	
16	TrackerEntry(String tag, EntryType type)		Tag, type	
17	TrackerEntry(Location loc)		loc	
18	getNullableFloat(Cursor cursor, String colName)	Float		

Table 57 - TrackerEntry

4.2.1.2.2.9 TrackDataHelper

Package	com.android.mts.mbtrack				
Class	TrackerProvider				
Attribute					
	No	Name	Type	Description	
	01	mFormatter	IFormatter		
	02	mContext	Context		
Method					
	No	Name	Return Type	Parameter	Description
	01	TrackDataHelper(Context context, IFormatter formatter)		Context, formatter	Create instance
	02	TrackDataHelper(Context context)		context	Create instance
	03	deleteAll()			Delete all tracker entries
	04	getNextOutput(Cursor cursor)	String	cursor	Helper method which converts row referenced by given cursor to a string output
	05	getOutputFooter()	String		Return output footer particular to

					the associated formatter
	06	getOutputHeader()	String		Return output header particular to the associated formatter
	07	query(string tag, int limit)	Cursor	tag, limit	Query tracker data filtering by given tag
	08	query(int limit)	Cursor	limit	Retrieves a cursor that starts at the last limit rows
	09	query(String tag)	Cursor	tag	Query tracker data, filtering by given tag
	10	writeEntry(TrackerEntry entry)		entry	Insert given TrackerEntry into content provinder
	11	writeEntry(Location loc, float distFromNetLoc)		loc, distFromNetLoc	Insert given location into tracker data
	12	writeEntry(String tag, String logMsg)		Tag, logMsg	Insert given log message into tracker data

Table 58 - TrackDataHelper

4.2.1.2.2.10 EnableConnectionService

Package	com.android.mts.service														
Class	EnableConnectionService														
Attribute	<table> <tr> <th>No</th><th>Name</th><th>Type</th><th>Description</th></tr> <tr> <td>01</td><td>URL</td><td>String</td><td></td></tr> <tr> <td>02</td><td>configData</td><td>ConfigDataSource</td><td></td></tr> </table>			No	Name	Type	Description	01	URL	String		02	configData	ConfigDataSource	
No	Name	Type	Description												
01	URL	String													
02	configData	ConfigDataSource													

Method				
	No	Name	Return Type	Parameter
	01	isOnline()	T/F	
	02	onBind()		
	03	onCreate()		
	04	onDestroy()		
	05	onStart(Intent intent, int startid)		Intent, startid
	06	getDangerousLocation()	List	
	07	getTimeDuaration	int	

Table 59 - EnableConnectionService

4.2.1.2.2.11 DataBaseManager

Package	com.android.mts.db		
Class	DataBaseManager		
Attribute			
	No	Name	Type
	01	mContext	Context
	02	mDataBase	SQLiteDatabase

Method					
	No	Name	Return Type	Parameter	Description
	01	DataBaseManager(Context context)		context	Constructor takes and keeps a reference of the passed context in order to access to the application assets and resources.
	02	select(String query)	Cursor	query	Return cursor with the results
	03	getDataTracking()	Array		
	04	getDataSMS()	Array		Get data from table sms
	05	getDataCallLog()	Array		Get data from table call log
	06	deleteAllDataTable(String tableName)		tableName	
	07	insert(String table, ContentValues values)		table, values	
	08	delete(String table, String where)		table, where	
	09	update(String table, ContentValues values, String where)		table, values, where	
	10	sqlCommand(String command)		command	Make a raw query
	11	close()			
	12	onCreate(SQLiteDatabase db)		db	
13	onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)		db, oldVersion, newVersion		

Table 60 - DataBaseManager

4.2.1.3 Call Logs

4.2.1.3.1 Class diagram

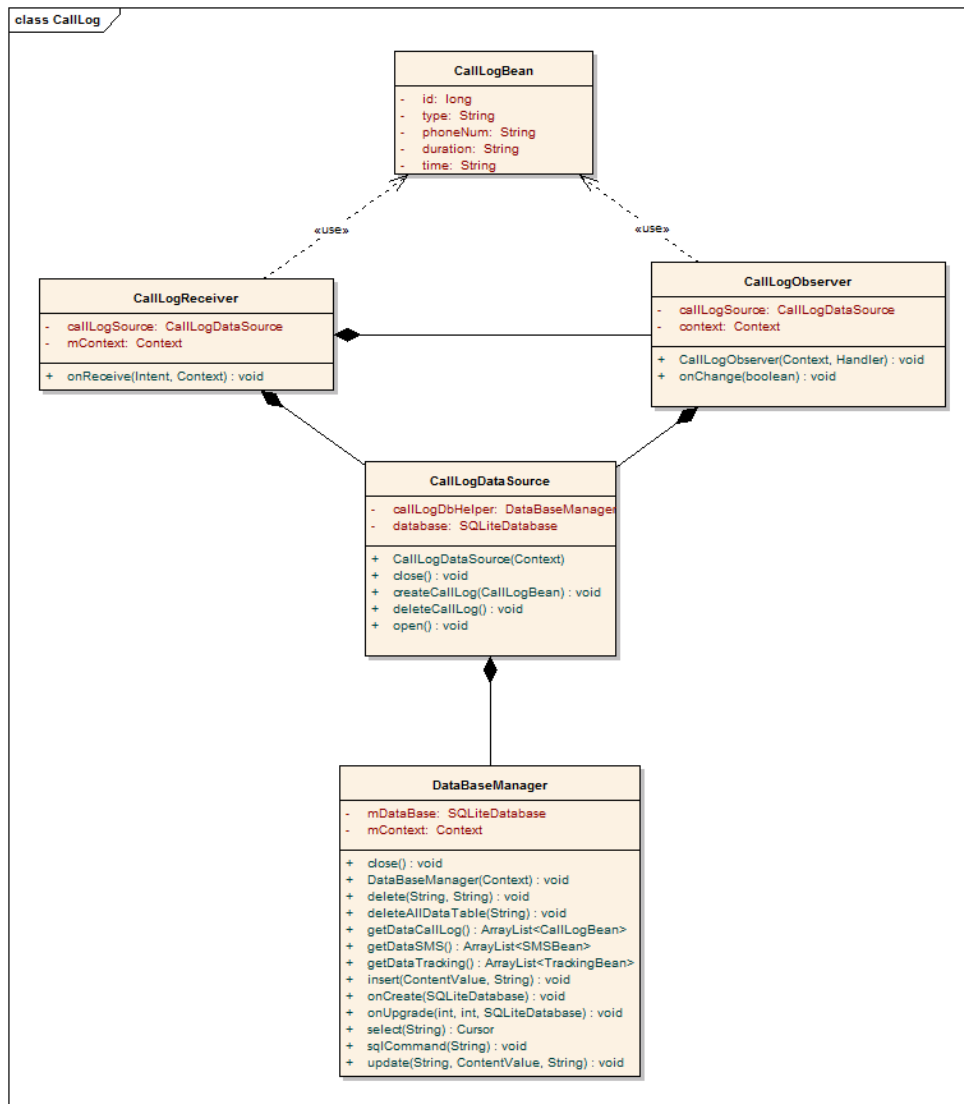


Figure 37 – Call Logs Class Diagram

4.2.1.3.2 Class explanation

4.2.1.3.2.1 CallLogBean

Package	com.android.mts.bean
Class	CallLogBean

Attribute				
	No	Name	Type	Description
	01	id	Long	
	02	type	String	
	03	phoneNum	String	
	04	duration	String	
	05	time	String	
Method				

Table 61 - CallLogBean

4.2.1.3.2.2 CallLogObserver

Package	com.android.mts.callog																		
Class	CallLogObserver																		
Attribute	<table><tr><th>No</th><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>01</td><td>callLogSource</td><td>CallLogDataSource</td><td></td></tr><tr><td>02</td><td>context</td><td>Context</td><td></td></tr></table>				No	Name	Type	Description	01	callLogSource	CallLogDataSource		02	context	Context				
No	Name	Type	Description																
01	callLogSource	CallLogDataSource																	
02	context	Context																	
Method	<table><tr><th>No</th><th>Name</th><th>Return Type</th><th>Parameter</th><th>Description</th></tr><tr><td>01</td><td>onChange(boolean selfChange)</td><td></td><td>selfChange</td><td></td></tr><tr><td>02</td><td>CallLogObserver(Handler handler, Context context)</td><td></td><td>Handler,context</td><td></td></tr></table>				No	Name	Return Type	Parameter	Description	01	onChange(boolean selfChange)		selfChange		02	CallLogObserver(Handler handler, Context context)		Handler,context	
No	Name	Return Type	Parameter	Description															
01	onChange(boolean selfChange)		selfChange																
02	CallLogObserver(Handler handler, Context context)		Handler,context																

Table 62 - CallLogObserver

4.2.1.3.2.3 CallLogDataSource

Package	com.android.mts.callog			
Class	CallLogDataSource			

Attribute

No	Name	Type	Description
01	callLogDbHelper	DataBaseManager	
02	Database	SQLiteDatabase	

Method

No	Name	Return Type	Parameter	Description
01	CallLogDataSource(Context context)		context	
02	close()			
03	open()			
04	deleteCallLog()			Delete call log
05	createCallLog(CallLogBean call)		call	Save call log into database

Table 63 - CallLogDataSource

4.2.1.3.2.4 DataBaseManager

Package	com.android.mts.db			
Class	DataBaseManager			
Attribute				
	No	Name	Type	Description
	01	mContext	Context	
	02	mDataBase	SQLiteDatabase	

Method

No	Name	Return Type	Parameter	Description
01	DataBaseManager(Context context)		context	Constructor takes and keeps a reference of the passed context in order to access to the application assets and resources.
02	select(String query)	Cursor	query	Return cursor with the results
03	getDataTracking()	Array		
04	getDataSMS()	Array		Get data from table sms
05	getDataCallLog()	Array		Get data from table call log
06	deleteAllDataTable(String tableName)		tableName	
07	insert(String table, ContentValues values)		table, values	
08	delete(String table, String where)		table, where	
09	update(String table, ContentValues values, String where)		table, values, where	
10	sqlCommand(String command)		command	Make a raw query
11	close()			
12	onCreate(SQLiteDatabase db)		db	
13	onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)		db, oldVersion, newVersion	

Table 64 - DataBaseManager

4.2.1.3.2.5 CallLogReceiver

Package	com.android.mts.callog			
Class	CallLogReceiver			
Attribute				
	No	Name	Type	Description
	01	callLogSource	CallLogDataSource	
Method	No	Name	Return Type	Parameter
	01	onReceive(Context context, Intent intent)		context, intent

Table 65– CallLogReceiver

4.2.1.4 Upload Service

4.2.1.4.1 Class Diagram

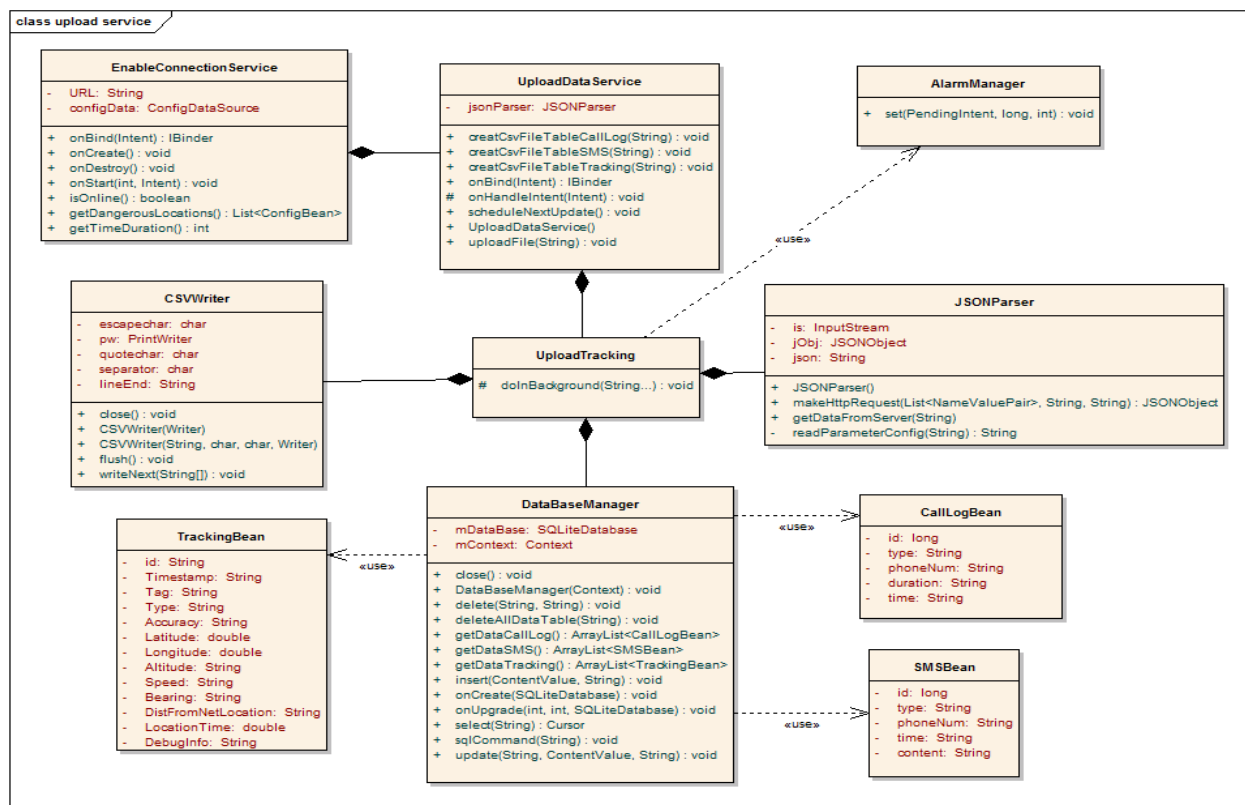


Figure 38 - Upload Service Class Diagram

4.2.1.4.2 Class Explanation

4.2.1.4.2.1 EnableConnectionService

Package	com.android.mts.service				
Class	EnableConnectionService				
Attribute					
	No	Name	Type	Description	
	01	URL	String		
	02	configData	ConfigDataSource		
Method					
	No	Name	Return Type	Parameter	Description
	01	isOnline()	T/F		
	02	onBind()			
	03	onCreate()			
	04	onDestroy()			
	05	onStart(Intent intent, int startid)		Intent, startid	
	06	getDangerousLocation()	List		
	07	getTimeDuaration	int		

Table 66 - EnableConectionService

4.2.1.4.2.2 UploadDataService

Package	com.android.mts.service													
Class	UploadDataService													
Attribute	<table><tr><th>No</th><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>01</td><td>jsonParser</td><td>JSONParser</td><td></td></tr></table>				No	Name	Type	Description	01	jsonParser	JSONParser			
No	Name	Type	Description											
01	jsonParser	JSONParser												
Method	<table><tr><th>No</th><th>Name</th><th>Return Type</th><th>Parameter</th><th>Description</th></tr><tr><td>01</td><td>createCsvFileTableC</td><td></td><td>filename</td><td></td></tr></table>				No	Name	Return Type	Parameter	Description	01	createCsvFileTableC		filename	
No	Name	Return Type	Parameter	Description										
01	createCsvFileTableC		filename											

		allLog(String filename)			
	02	createCsvFileTableSMS(String filename)		filename	
	03	createCsvFileTableTracking(String filename)		filename	
	04	onBind(Intent intent)	IBinder	intent	
	05	onHandleIntent(Intent intent)		intent	
	06	scheduleNextUpdate()			
	07	UploadDataService()			
	08	uploadFile(String tableName)		tableName	

Table 67 - UploadDataService

4.2.1.4.2.3 UploadTracking

Package	com.android.mts.service													
Class	UploadTracking													
Attribute														
Method	<table> <tr> <th>No</th><th>Name</th><th>Return Type</th><th>Parameter</th><th>Description</th></tr> <tr> <td>01</td><td>doInBackground(String... args)</td><td></td><td></td><td>Upload data</td></tr> </table>				No	Name	Return Type	Parameter	Description	01	doInBackground(String... args)			Upload data
No	Name	Return Type	Parameter	Description										
01	doInBackground(String... args)			Upload data										

Table 68 - UploadTracking

4.2.1.4.2.4 CSVWriter

Package	com.android.mts.utils
Class	CSVWriter

Attribute					
	No	Name	Type	Description	
	01	escapechar	Char		
	02	lineEnd	String		
	03	pw	PrintWriter		
	04	quotechar	Char		
	05	separator	Char		
Method					
	No	Name	Return Type	Parameter	Description
	01	close()			Close the underlying stream writer flushing any buffered content
	02	CSVWriter(Writer writer)		writer	Constructs CSVWriter using a comma for the separator
	03	CSVWriter(Writer writer, char separator, char quotechar, char escapechar, String lineEnd)		Writer, separator, quotechar, escapechar, lineEnd	Constructs CSVWriter with supplied separator, quote char, escape char and line ending
	04	flush()			Flush underlying stream to writer
	05	writeNext(String[] nextLine)		nextLine	Writes the next line to the file

Table 69 - CSVWriter

4.2.1.4.2.5 JSONParser

Package	com.android.mts.utils			
Class	JSONParser			
Attribute				
	No	Name	Type	Description
	01	is	InputStream	
	02	jObj	JSONObject	
	03	Json	String	

Method	No	Name	Return Type	Parameter	Description
	01	JSONParser()			Constructor
	02	makeHttpRequest(String url, String method, List<NameValuePair> params)	JSONObject		Get json from url
	03	getDataFromServer(String url)	JSONArray	url	Get configuration data from server
	04	readParameterConfig(String url)	String	url	

Table 70 - JSONParser

4.2.1.4.2.6 DatabaseManager

Package	com.android.mts.db				
Class	DataBaseManager				
Attribute					
	No	Name	Type	Description	
	01	mContext	Context		
	02	mDataBase	SQLiteDatabase		
Method					
	No	Name	Return Type	Parameter	Description
	01	DataBaseManager(Context context)		context	Constructor takes and keeps a reference of the past context in order to access to the application assets and resources.
	02	select(String query)	Cursor	query	Return cursor with the results
	03	getDataTracking()	Array		
	04	getDataSMS()	Array		Get data from

					table sms
05	getDataCallLog()	Array			Get data from table call log
06	deleteAllDataTable(String tableName)		tableName		
07	insert(String table, ContentValues values)		table, values		
08	delete(String table, String where)		table, where		
09	update(String table, ContentValues values, String where)		table, values, where		
10	sqlCommand(String command)		command		Make a raw query
11	close()				
12	onCreate(SQLiteDatabase db)		db		
13	onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)		db, oldVersion, newVersion		

Table 71 - DatabaseManager

4.2.1.4.2.7 CallLogBean

Package	com.android.mts.bean			
Class	CallLogBean			
Attribute				
	No	Name	Type	Description
	01	id	Long	
	02	type	String	
	03	phoneNum	String	
	04	duration	String	
	05	time	String	
Method				

Table 72 - CallLogBean

4.2.1.4.2.8 TrackingBean

Package	com.android.mts.bean			
Class	TrackingBean			

Attribute			
	01	Accuracy	String
	02	Altitude	String
	03	Bearing	String
	04	DebugInfo	String
	05	DistFromNetLocation	String
	06	id	String
	07	Latitude	Double
	08	LocationTime	Double
	09	Longitude	Double
	10	Speed	String
	11	Tag	String
	12	Timestamp	String
	13	Type	String
Method			

Table 73 - TrackingBean

4.2.1.4.2.9 AlarmManager

Package	Android class			
Class	AlarmManager			
Attribute				
Method				
	No	Name	Return Type	Parameter
	01	Set(PendingIntent operation , long triggerAtMillis, int type)		operation, triggerAtmillis, type

Table 74 - AlarmManager

4.2.2 Server

4.2.2.1 Account

4.2.2.1.1 Class diagram

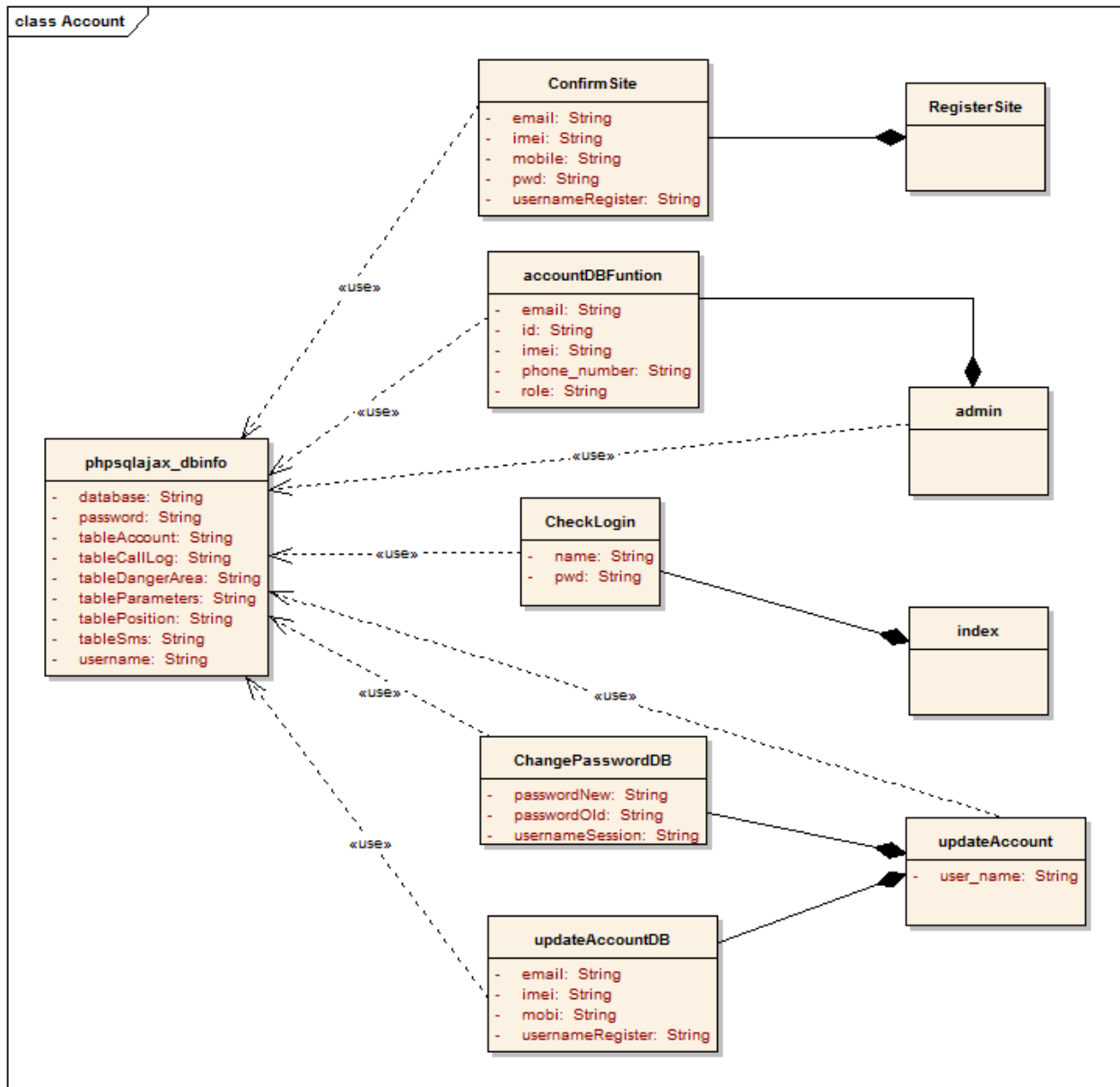


Figure 39 - Account Class Diagram

4.2.2.1.2 Class explanation

4.2.2.1.2.1 ConfirmSite

Package	
---------	--

Class	ConfirmSite			
Attribute				
	No	Name	Type	Description
	01	email	String	
	02	imei	String	
	03	mobile	String	
	04	pwd	String	
	05	usernameRegister	String	
Method				

Table 75 - ConfirmSite

4.2.2.1.2.2 AccountDBFuntion

Package	Common.DatabaseCommon			
Class	accountDBFuntion			
Attribute				
	No	Name	Type	Description
	01	email	String	
	02	id	String	
	03	imei	String	
	04	Phone_number	String	
	05	role	String	
Method				

Table 76 - AccountDBFunction

4.2.2.1.2.3 CheckLogin

Package	Business.Login															
Class	CheckLogin															
Attribute																
	<table><tr><th>No</th><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>01</td><td>name</td><td>String</td><td></td></tr><tr><td>02</td><td>pwd</td><td>String</td><td></td></tr></table>				No	Name	Type	Description	01	name	String		02	pwd	String	
	No	Name	Type	Description												
01	name	String														
02	pwd	String														
Method																

Table 77 - CheckLogin

4.2.2.1.2.4 ChangePasswordDB

Package	Business.UpdateAccount																		
Class	ChangePasswordDB																		
Attribute	<table> <tr> <th>No</th><th>Name</th><th>Type</th><th>Description</th></tr> <tr> <td>01</td><td>passwordNew</td><td>String</td><td></td></tr> <tr> <td>02</td><td>passwordOld</td><td>String</td><td></td></tr> <tr> <td>03</td><td>usernameSession</td><td>String</td><td></td></tr> </table>			No	Name	Type	Description	01	passwordNew	String		02	passwordOld	String		03	usernameSession	String	
No	Name	Type	Description																
01	passwordNew	String																	
02	passwordOld	String																	
03	usernameSession	String																	
Method																			

Table 78 - ChangePasswordDB

4.2.2.1.2.5 UpdateAccountDB

Package	Business.UpdateAccount																						
Class	updateAccountDB																						
Attribute	<table> <tr> <th>No</th><th>Name</th><th>Type</th><th>Description</th></tr> <tr> <td>01</td><td>email</td><td>String</td><td></td></tr> <tr> <td>02</td><td>imei</td><td>String</td><td></td></tr> <tr> <td>03</td><td>mobi</td><td>String</td><td></td></tr> <tr> <td>04</td><td>usernameRegister</td><td>String</td><td></td></tr> </table>			No	Name	Type	Description	01	email	String		02	imei	String		03	mobi	String		04	usernameRegister	String	
No	Name	Type	Description																				
01	email	String																					
02	imei	String																					
03	mobi	String																					
04	usernameRegister	String																					
Method																							

Table 79 - UpdateAccountDB

4.2.2.1.2.6 UpdateAccount

Package	Business.UpdateAccount										
Class	updateAccount										
Attribute	<table> <tr> <th>No</th><th>Name</th><th>Type</th><th>Description</th></tr> <tr> <td>01</td><td>updateAccount</td><td>String</td><td></td></tr> </table>			No	Name	Type	Description	01	updateAccount	String	
No	Name	Type	Description								
01	updateAccount	String									
Method											

Table 80 - UpdateAccount

4.2.2.1.2.7 Phpsqlajax_dbinfo

Package	Database																																											
Class	phpsqlajax_dbinfo																																											
Attribute	<table><tr><th>No</th><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>01</td><td>database</td><td>String</td><td></td></tr><tr><td>02</td><td>password</td><td>String</td><td></td></tr><tr><td>03</td><td>tableAccount</td><td>String</td><td></td></tr><tr><td>04</td><td>tableCallLog</td><td>String</td><td></td></tr><tr><td>05</td><td>tableDangerArea</td><td>String</td><td></td></tr><tr><td>06</td><td>tableParameters</td><td>String</td><td></td></tr><tr><td>07</td><td>tablePosition</td><td>String</td><td></td></tr><tr><td>08</td><td>tableSms</td><td>String</td><td></td></tr><tr><td>09</td><td>username</td><td>String</td><td></td></tr></table>				No	Name	Type	Description	01	database	String		02	password	String		03	tableAccount	String		04	tableCallLog	String		05	tableDangerArea	String		06	tableParameters	String		07	tablePosition	String		08	tableSms	String		09	username	String	
	No	Name	Type	Description																																								
	01	database	String																																									
	02	password	String																																									
	03	tableAccount	String																																									
	04	tableCallLog	String																																									
	05	tableDangerArea	String																																									
	06	tableParameters	String																																									
	07	tablePosition	String																																									
	08	tableSms	String																																									
09	username	String																																										
Method																																												

Table 81 - Phpsqlajax_dbinfo

4.2.2.2 Danger Area

4.2.2.2.1 Class diagram

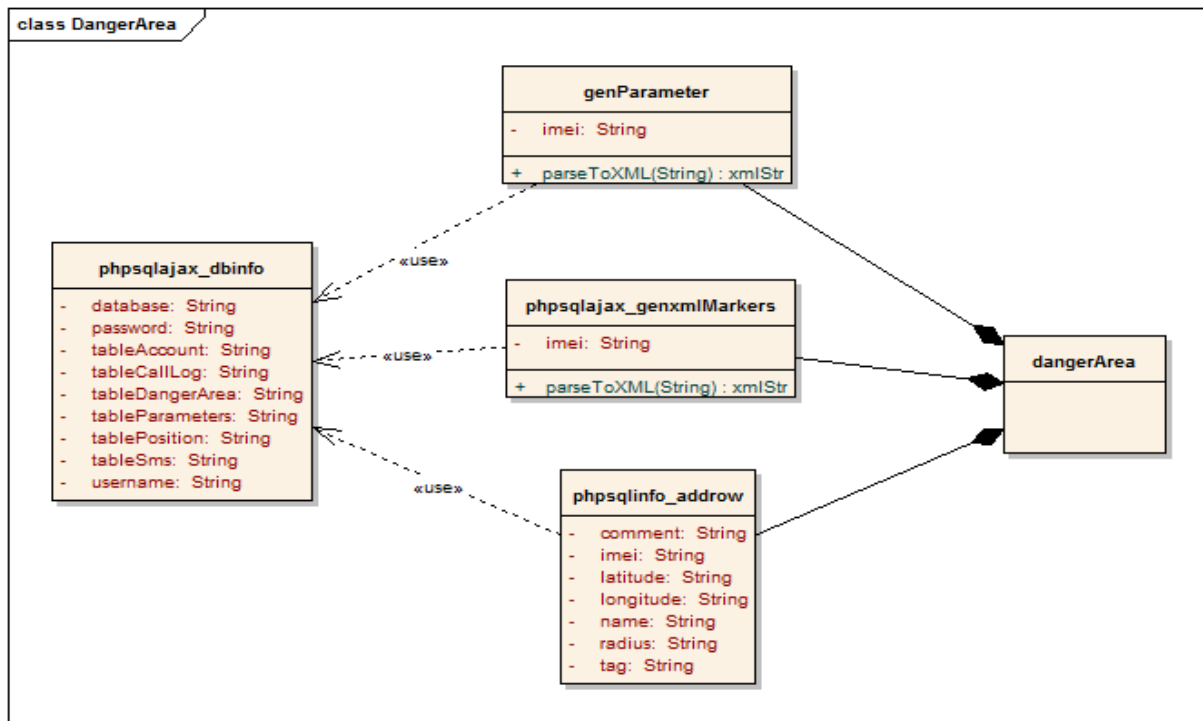


Figure 40 - DangerArea Class Diagram

4.2.2.2.2 Class explanation

4.2.2.2.2.1 GenParameter

Package	Business.Marker				
Class	genParameter				
Attribute					
	No	Name	Type	Description	
	01	imei	String		
Method					
	No	Name	Return Type	Parameter	Description
	01	parseToXML(String htmlStr)	xmlStr	htmlStr	

Table 82 - GenParameter

4.2.2.2.2.2 Phpsqlajax_genxmlMarkers

Package	Business.Marker														
Clas	phpsqlajax_genxmlMarkers														
Attribute	<table><tr><th>No</th><th>Name</th><th>Type</th><th colspan="2">Description</th></tr><tr><td>01</td><td>imei</td><td>String</td><td colspan="2"></td></tr></table>					No	Name	Type	Description		01	imei	String		
No	Name	Type	Description												
01	imei	String													
Method	<table><tr><th>No</th><th>Name</th><th>Return Type</th><th>Parameter</th><th>Description</th></tr><tr><td>01</td><td>parseToXML(String htmlStr)</td><td>xmlStr</td><td>htmlStr</td><td></td></tr></table>					No	Name	Return Type	Parameter	Description	01	parseToXML(String htmlStr)	xmlStr	htmlStr	
No	Name	Return Type	Parameter	Description											
01	parseToXML(String htmlStr)	xmlStr	htmlStr												

Table 83 - Phpsqlajax_genxmlMarkers

4.2.2.2.2.3 phpsqlinfo_addrow

Package	Database				
Class	phpsqlinfo_addrow				

Attribute				
	No	Name	Type	Description
	01	comment	String	
	02	imei	String	
	03	latitude	String	
	04	longitude	String	
	05	name	String	
	06	radius	String	
	07	tag	String	
Method				

Table 84 - Phpsqlinfo_addrow

4.2.2.2.2.4 phpsqlajax_dbinfo

Package	Database			
Class	phpsqlajax_dbinfo			
Attribute				
	No	Name	Type	Description
	01	database	String	
	02	password	String	
	03	tableAccount	String	
	04	tableCallLog	String	
	05	tableDangerArea	String	
	06	tableParameters	String	
	07	tablePosition	String	
	08	tableSms	String	
	09	username	String	
Method				

Table 85 - Phpsqlajax_dbinfo

4.2.2.3 Service

4.2.2.3.1 Class diagram

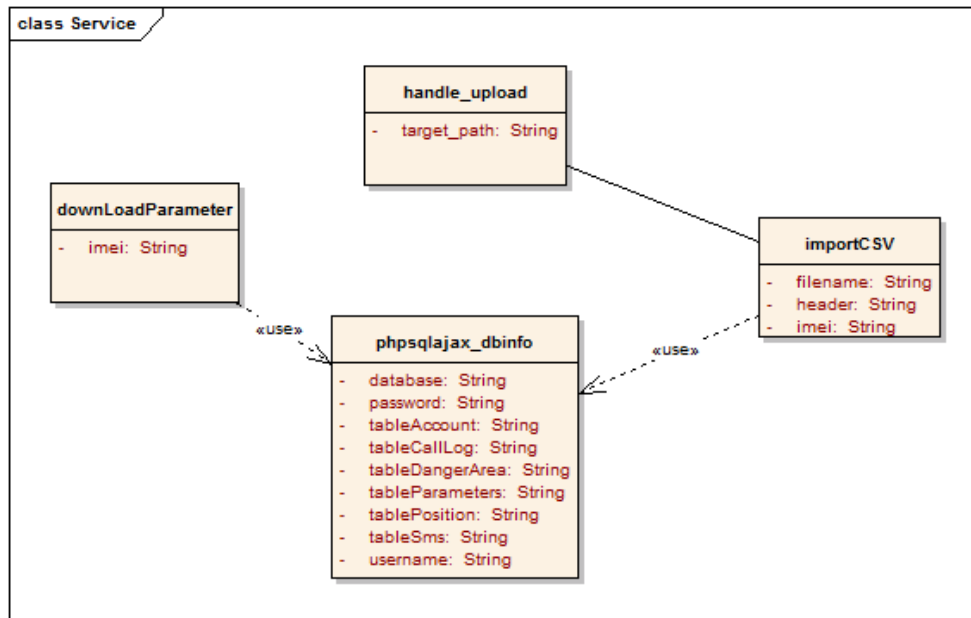


Figure 41 - Service Class Diagram

4.2.2.3.2 Class explanation

4.2.2.3.2.1 Handle_upload

Package	Business.Service.Upload			
Class	handle_upload			
Attribute				
	No	Name	Type	Description
	01	target_path	String	
Method				

Table 86 - Handle_Upload

4.2.2.3.2.2 DownloadParameter

Package	Business.Service.Download			
Class	downloadParameter			

Attribute				
	No	Name	Type	Description
	01	imei	String	
Method				

Table 87 - DownLoadParameter

4.2.2.3.2.3 phpsqlajax_dbinfo

Package	Database			
Class	phpsqlajax_dbinfo			
Attribute	No	Name	Type	Description
	01	database	String	
	02	password	String	
	03	tableAccount	String	
	04	tableCallLog	String	
	05	tableDangerArea	String	
	06	tableParameters	String	
	07	tablePosition	String	
	08	tableSms	String	
	09	username	String	
Method				

Table 88 - Phpsqlajax_dbinfo

4.2.2.3.2.4 ImportCSV

Package	Business.Service.Upload			
Class	importCSV			
Attribute	No	Name	Type	Description
	01	filename	String	
	02	header	String	
	03	imei	String	
Method				

Table 89 - ImportCSV

4.2.2.4 Show History

4.2.2.4.1 Class diagram

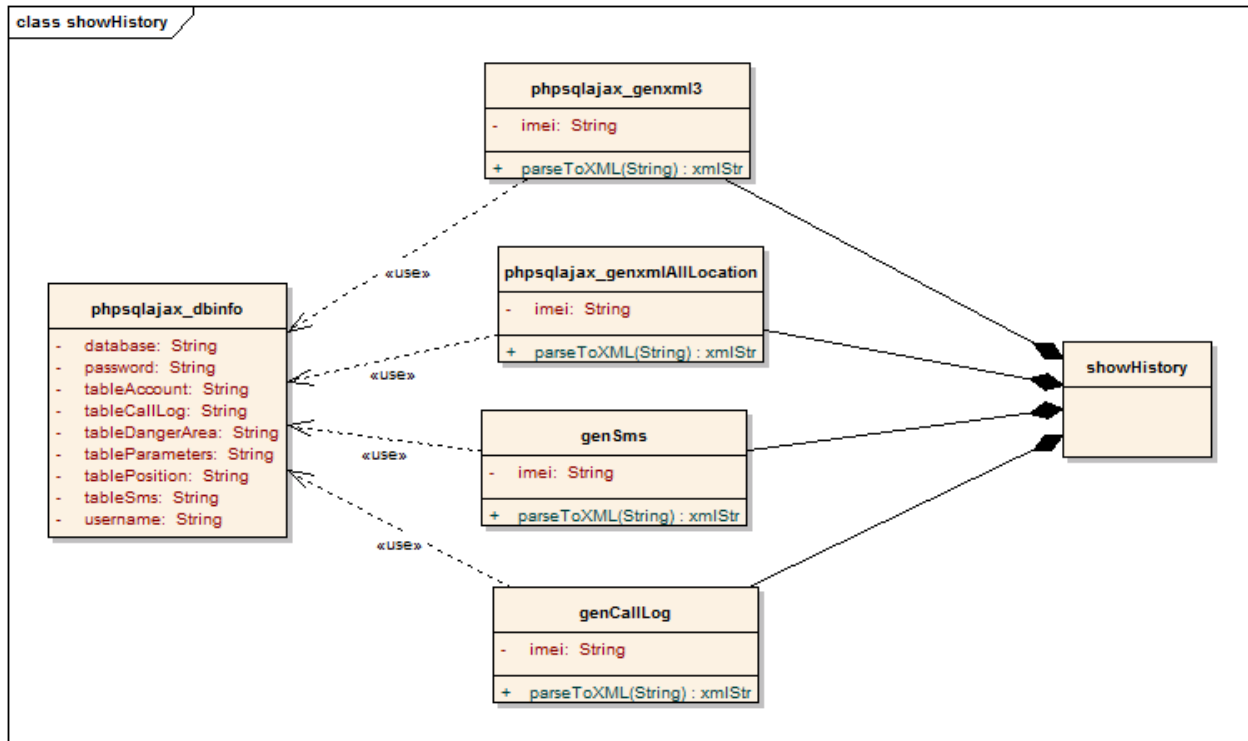


Figure 42- ShowHistory Class Diagram

4.2.2.4.2 Class explanation

4.2.2.4.2.1 Phpsqlajax_genxml3

Package	Business.History.Location				
Class	phpsplajax_genxml3				
Attribute					
	No	Name	Type	Description	
	01	imei	String		
Method					
	01	parseToXML(String htmlStr)	xmlStr	htmlStr	

Table 90 - Phpsqlajax_genxml3

4.2.2.4.2.2 Phpsqlajax_genxmlAllLocation

Package	Business.History.Location				
Class	phpsplajax_genxmlAllLocation				
Attribute					
	No	Name	Type	Description	
	01	imei	String		
Method	No	Name	Return Type	Parameter	Description
	01	parseToXML(String htmlStr)	xmlStr	htmlStr	

Table 91 - Phpsqlajax_genxmlAllLocation

4.2.2.4.2.3 GenSms

Package	Business.History.SMS				
Class	genSms				
Attribute					
	No	Name	Type	Description	
	01	imei	String		
Method	No	Name	Return Type	Parameter	Description
	01	parseToXML(String htmlStr)	xmlStr	htmlStr	

Table 92 - GenSMS

4.2.2.4.2.4 GenCallLog

Package	Business.History.CallLog				
Class	genCallLog				
Attribute					
	No	Name	Type	Description	
	01	imei	String		
Method	No	Name	Return Type	Parameter	Description
	01	parseToXML(String htmlStr)	xmlStr	htmlStr	

Table 93 - GenCallLog

4.2.2.4.2.5 phpsqlajax_dbinfo

Package	Database																																										
Class	phpsqlajax_dbinfo																																										
Attribute	<table border="1"> <thead> <tr> <th>No</th><th>Name</th><th>Type</th><th>Description</th></tr> </thead> <tbody> <tr><td>01</td><td>database</td><td>String</td><td></td></tr> <tr><td>02</td><td>password</td><td>String</td><td></td></tr> <tr><td>03</td><td>tableAccount</td><td>String</td><td></td></tr> <tr><td>04</td><td>tableCallLog</td><td>String</td><td></td></tr> <tr><td>05</td><td>tableDangerArea</td><td>String</td><td></td></tr> <tr><td>06</td><td>tableParameters</td><td>String</td><td></td></tr> <tr><td>07</td><td>tablePosition</td><td>String</td><td></td></tr> <tr><td>08</td><td>tableSms</td><td>String</td><td></td></tr> <tr><td>09</td><td>username</td><td>String</td><td></td></tr> </tbody> </table>			No	Name	Type	Description	01	database	String		02	password	String		03	tableAccount	String		04	tableCallLog	String		05	tableDangerArea	String		06	tableParameters	String		07	tablePosition	String		08	tableSms	String		09	username	String	
No	Name	Type	Description																																								
01	database	String																																									
02	password	String																																									
03	tableAccount	String																																									
04	tableCallLog	String																																									
05	tableDangerArea	String																																									
06	tableParameters	String																																									
07	tablePosition	String																																									
08	tableSms	String																																									
09	username	String																																									
Method																																											

Table 94 - Phpsqlajax_dbinfo

4.3 Sequence Diagram

4.3.1 Mobile

4.3.1.1 SMS

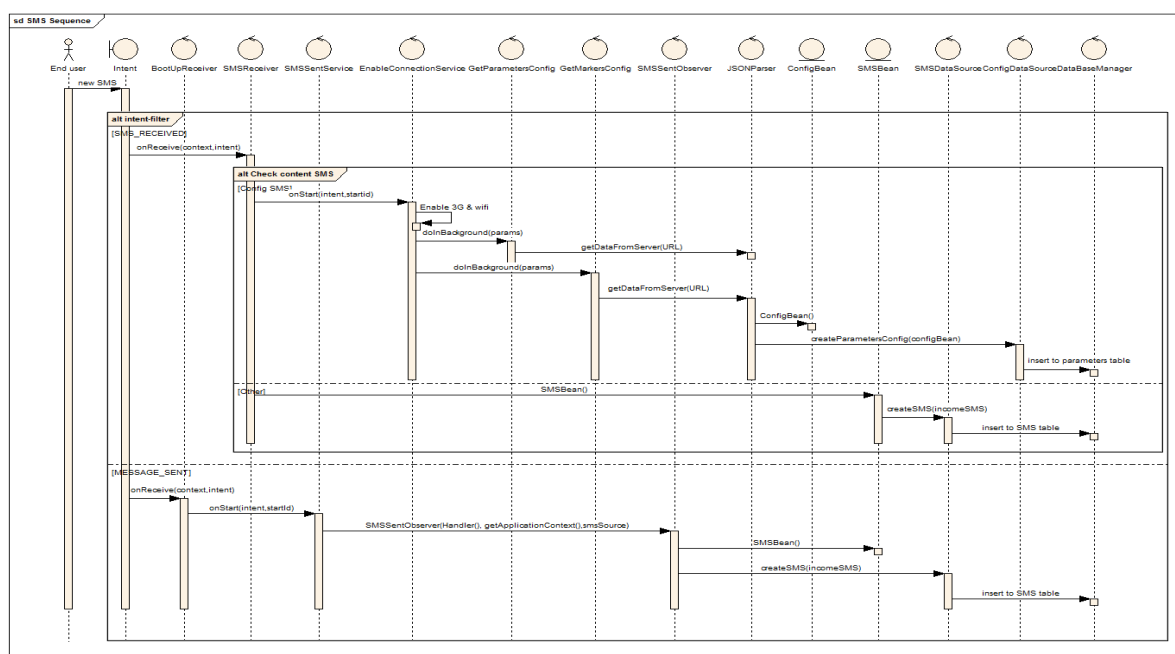


Figure 43 - SMS Sequence Diagram

No	Method Name	Description
01	onReceive(Context context, Intent intent)	-Description: When receiving new sms, call function:SMSReceiver.onReceive(context,intent). -Parameter: context, intent -Return: N/A
02	onStart(Intent intent, int startId)	-Description: If received sms is configuration sms, call function: EnableConnectionService.onStart(intent,startId) to automatically enable network and hide this sms. -Parameter: intent, 120startId -Return: N/A
03	Enable 3G & wifi	-Description: Automatically turned on 3G & wifi.. -Parameter: N/A -Return: N/A
04	doInBackground(String...params)	-Description: get configuration parameters/ list of dangerous positions from server. -Parameter: params -Return: String
05	getDataFromServer(String URL)	-Description: get data from server. -Parameter: URL -Return: N/A
06	ConfigBean()	-Description: Constructor function to save to parameters table of mobile database. -Parameter: -Return:
07	createParametersConfig(ConfigBean config)	-Description: Set value to ConfigBean object. -Parameter: config -Return: N/A
08	insert(String table, String nullColumnHack, ContentValues values)	- Description: Insert to parameters table. -Parameter: table,nullColumnHack,values -Return: N/A
09	SMSBean()	-Description: Constructor function to save to sms table of mobile database. -Parameter: N/A -Return: N/A
10	createSMS(SMSBean sms)	-Description: Save all data to sms table of mobile database -Parameter: sms -Return: N/A
11	insert(String table, String nullColumnHack, ContentValues values)	- Description: Insert to sms table. -Parameter: table,nullColumnHack,values -Return: N/A

12	onReceive(Context context, Intent intent)	-Description: When end users sent sms, call function: BootupReceiver.onReceive(context,intent). -Parameter: context,intent -Return: N/A
13	onStart(Intent intent, int startId)	-Description: Get sms content to save to mobile database. -Parameter: intent, startId -Return: N/A
14	SMSSentObserver(Handler hander, Context ctx , SMSDataSource smsSource)	-Description: Constructor to call SMSSentObserver service. -Parameter: handler, ctx, smsSource -Return: N/A

Table 95 - SMS Sequence Description

4.3.1.2 MobileTrack

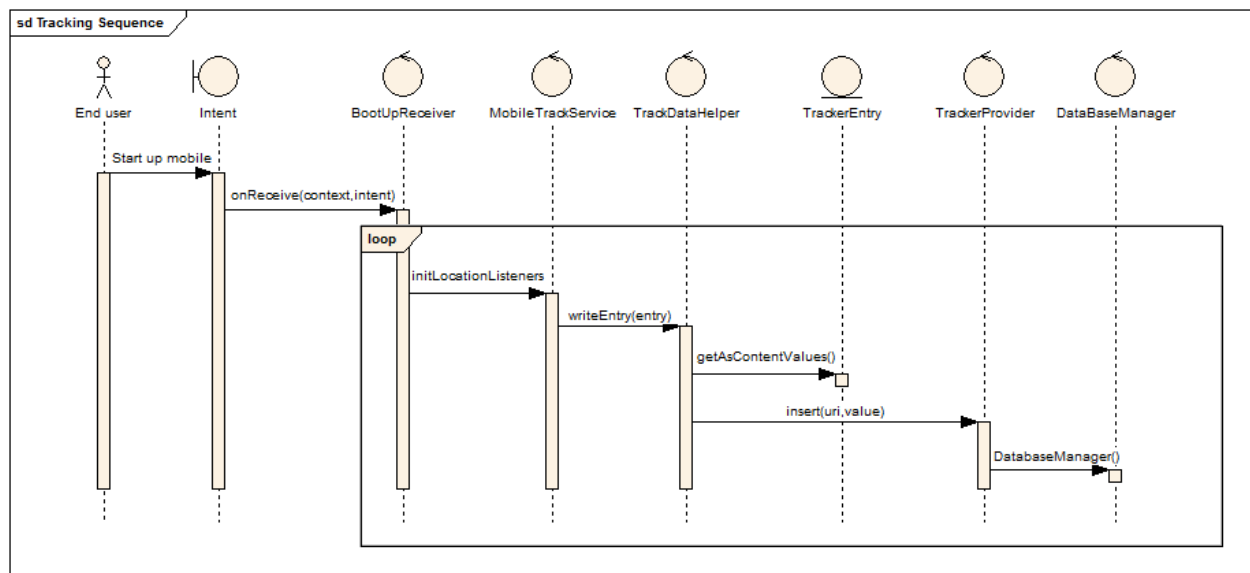


Figure 44 - Mobile Track Sequence Diagram

No	Method Name	Description
01	onReceive(Context context, Intent intent)	-Description: When start up mobile, call function : BootUpReceiver.onReceive(context,intent). -Parameter: context, intent -Return: N/A
02	initLocationListeners()	-Description: Initialize LocationListeners service to tracking position. -Parameter: N/A -Return: N/A
03	writeEntry(TrackerEntry	-Description: Insert data into TrackerProvider object.

	entry)	-Parameter: entry -Return: N/A
04	getAsContentValues()	-Description: Set data into TrackerEntry object. -Parameter: -Return: ContentValues
05	insert(Uri uri, ContentValues values)	-Description: Insert to tracking table. -Parameter: uri, values -Return: Uri
06	DatabaseManager()	-Description: Constructor to connect database. -Parameter: N/A -Return: N/A

Table 96 – Mobile Track Sequence Description

4.3.1.3 Call Logs

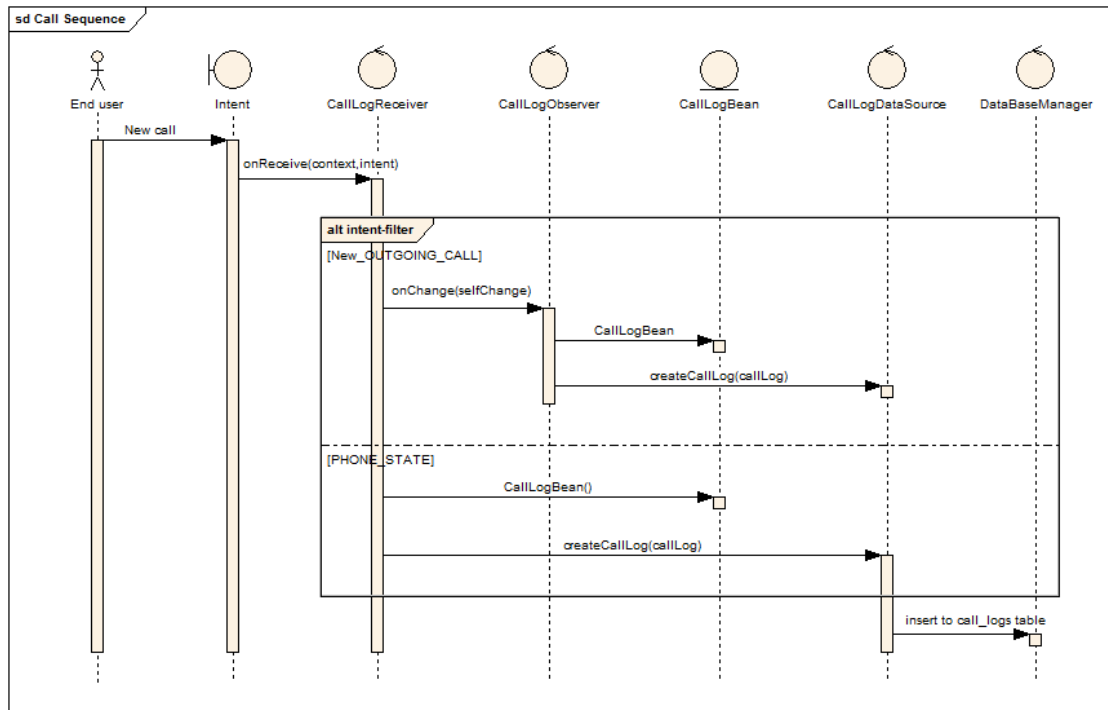


Figure 45 - Call Logs Sequence Diagram

No	Method Name	Description
01	onReceive(Context context, Intent intent)	-Description: When have any new call, call this function. -Parameter: context, intent -Return: N/A
02	onChange(boolean selfChange)	-Description: If new call is outgoing call, call this function. -Parameter: selfChange -Return: N/A
03	CallLogBean()	-Description: Constructor function to save to call_logs table of mobile database.

		-Parameter: N/A -Return: N/A
04	createCallLog(CallLogBean call)	-Description: Save all data to call_logs table of mobile database -Parameter: call -Return: N/A
05	insertOrThrow(String table, String nullColumnHack, ContentValues values)	-Description: Insert to call_logs table -Parameter: table,nullColumnHack,values -Return: N/A

Table 97 – Call Logs Sequence Description

4.3.1.4 UploadData

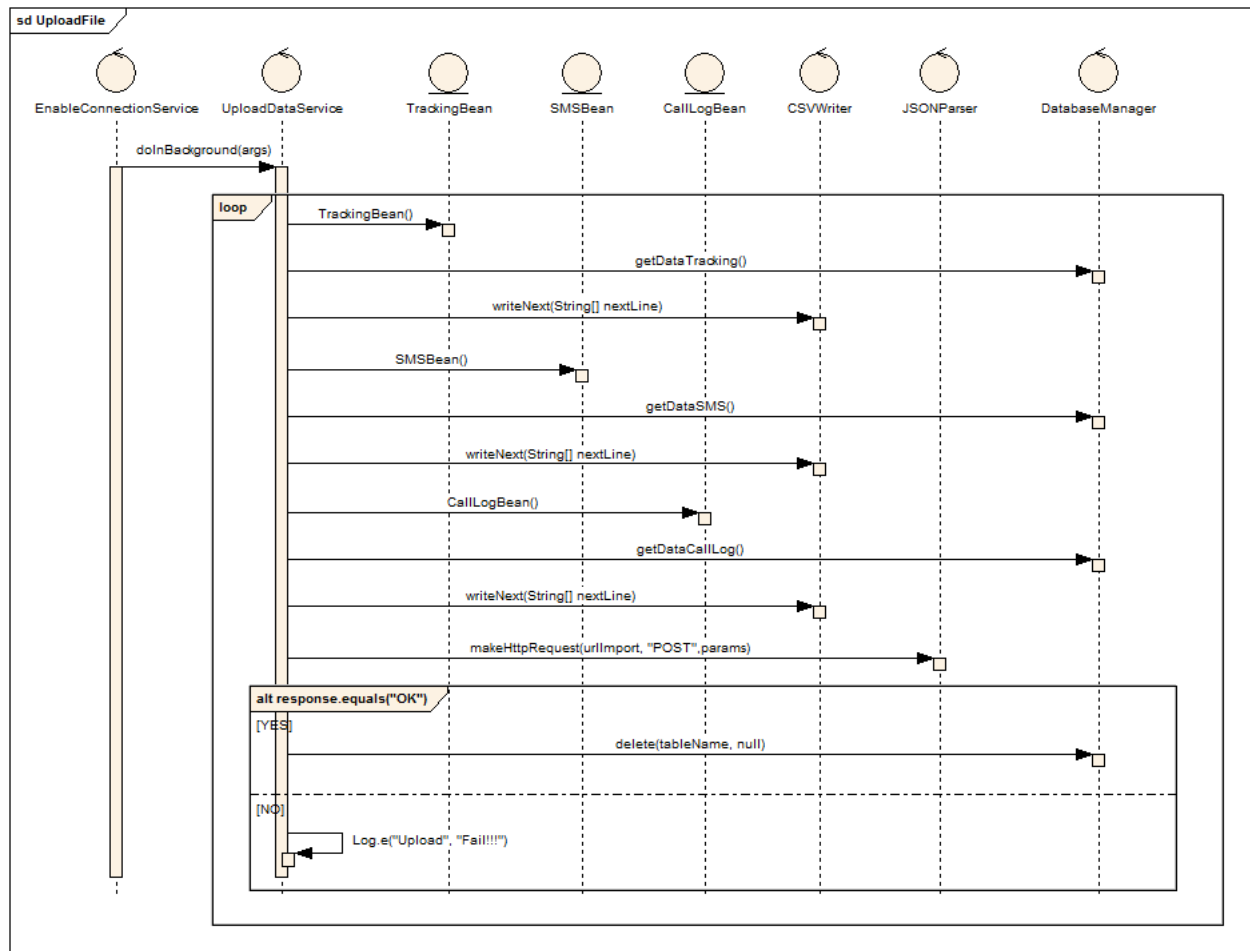


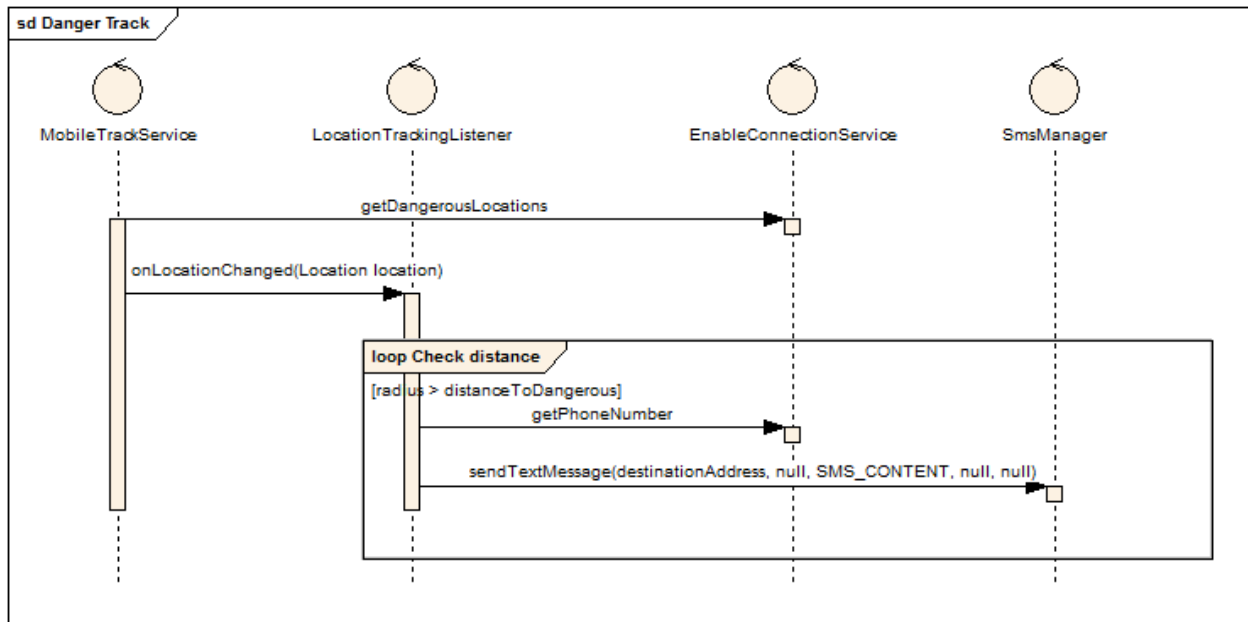
Figure 46 - Upload Data Sequence Diagram

No	Method Name	Description
01	doInBackground(String...args)	-Description: After configuration time, call this function to upload data from mobile to server. -Parameter: args -Return: N/A

02	TrackingBean()	-Description: Constructor function to save to positions object. -Parameter: N/A -Return: N/A
03	getDataTracking()	-Description: Get data from mobile and save to TrackingBean object -Parameter: N/A -Return: ArrayList<TrackingBean>
04	writeNext(String[] nextLine)	-Description: Write data to CSV file to upload. -Parameter: nextLine -Return: N/A
05	smsBean()	-Description: Constructor function to save to sms object. -Parameter: N/A -Return: N/A
06	getDataSMS()	-Description: Get data from mobile and save to SMSBean object. -Parameter: N/A -Return: ArrayList<SMSBean>
09	CallLogBean()	-Description: Constructor function to save to CallLogBean object. -Parameter: N/A -Return: N/A
10	getDataCallLog()	-Description: Get data from mobile and save to CallLogBean object. -Parameter: N/A -Return: ArrayList<CallLogBean>
12	makeHttpRequest(String url, String method, List<NameValuePair> params)	-Description: Get response from server. -Parameter: url, method, params -Return: JSONObject
13	delete(String tableName,String where)	-Description: If upload successfully, delete data that saved in mobile database. -Parameter: tableName,where -Return: N/A
14	Log.e ("Upload", "Fail!")	-Description: If upload fail, log event. -Parameter: string, string -Return: N/A

Table 98 - Upload Data Sequence Description

4.3.1.5 Danger Track



No	Method Name	Description
01	getDangerousLocations	-Description: get list of danger positions. -Parameter: N/A -Return: N/A
02	onLocationChanged(Location location)	-Description: Process when location changed. -Parameter: location -Return: N/A
03	getPhoneNumber	-Description: get phone number of supervisor -Parameter: N/A -Return: N/A
04	sendTextMessage(DestinationAddress, null, SMS_CONTENT, null, null)	-Description: Send sms to supervisor.. -Parameter: N/A -Return: N/A

Table 99 - Danger Track Sequence Description

4.3.2 Server

4.3.2.1 Danger Area Sequence

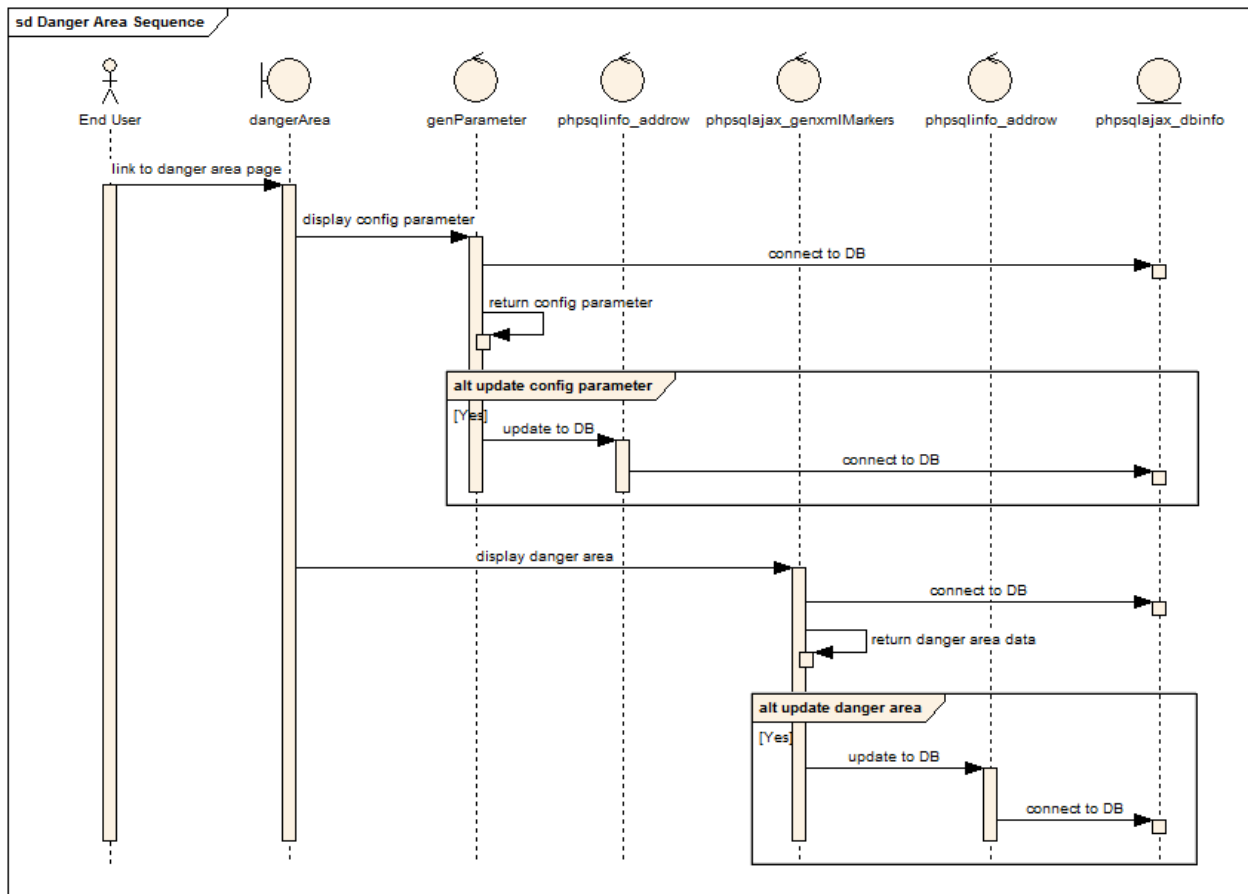


Figure 47 - Danger Area Sequence Diagram

No	Method Name	Description
01	Link to danger area page	-Description: Ways to go to this page: + After login successfully, go to this page immediately +Click to History Tab on Menu Tab -Parameter: N/A -Return: N/A
02	Display configuration parameter	-Description: Configuration parameter includes configuration time that end users defined. -Parameter: N/A -Return: N/A
03	Connect database	-Description: Connect database. -Parameter: N/A -Return: N/A
04	Update to database	-Description: If end users changed data, update these

		changes to database. Data can changed includes: +Time Duration +Danger Area -Parameter: N/A -Return: N/A
05	Display danger area	-Description: Display danger area on map. -Parameter: N/A -Return: N/A

Table 100- Danger Area Sequence Description

4.3.2.2 Login

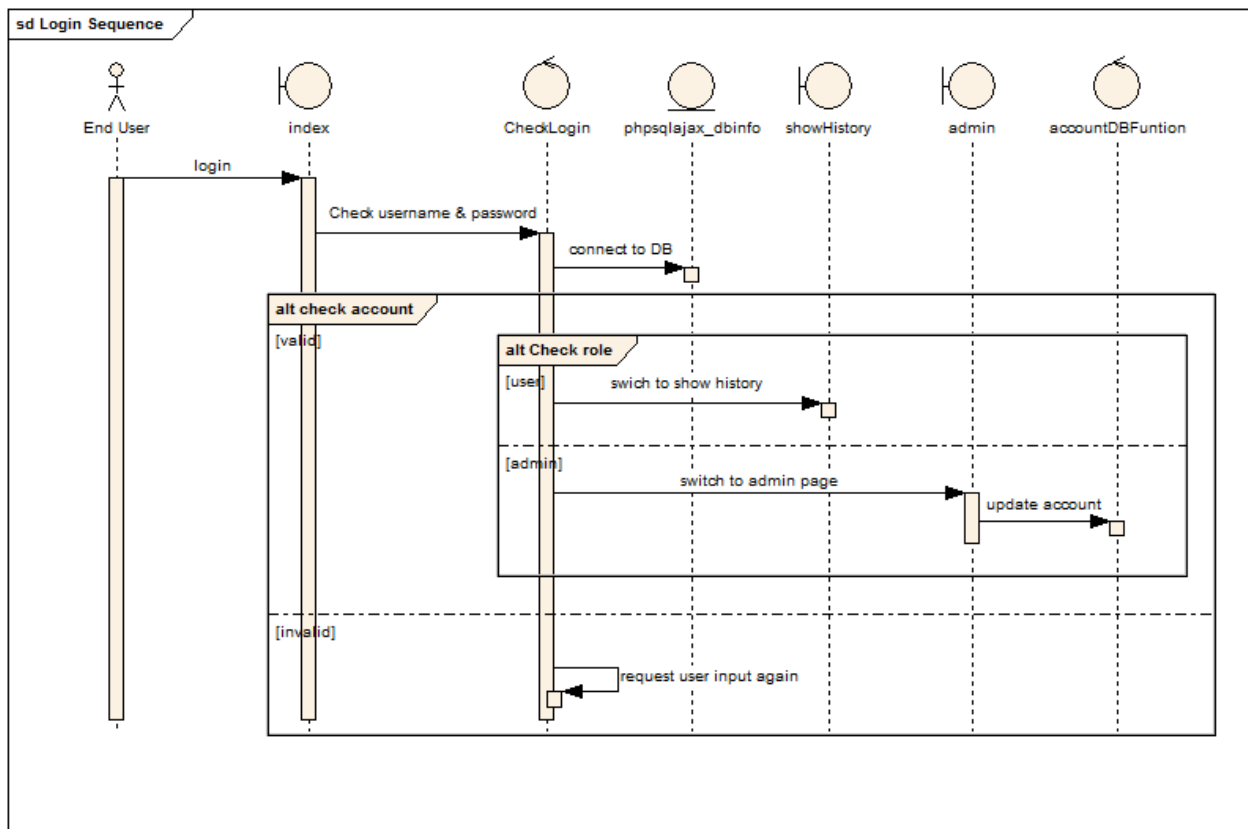


Figure 48 - Login Sequence Diagram

No	Method Name	Description
01	Login	-Description: Login to use system. -Parameter: user name, password -Return: N/A
02	Check user name & password	-Description: Check user name & password to access. -Parameter: user name, password -Return: T/F
03	Connect to database	-Description: Connect database -Parameter: N/A

		-Return: N/A
04	Switch to show history	-Description: Login by end user role, go to history page. -Parameter: N/A -Return: N/A
05	Switch to admin page	-Description: Login by admin role, go to admin page. -Parameter: N/A -Return: N/A
06	Update account	-Description: As admin role, if account information has changed, update to database. -Parameter: N/A -Return: N/A
07	Request user input again	-Description: Re-input when input wrong. -Parameter: N/A -Return: N/A

Table 101 - Login Sequence Description

4.3.2.3 Register

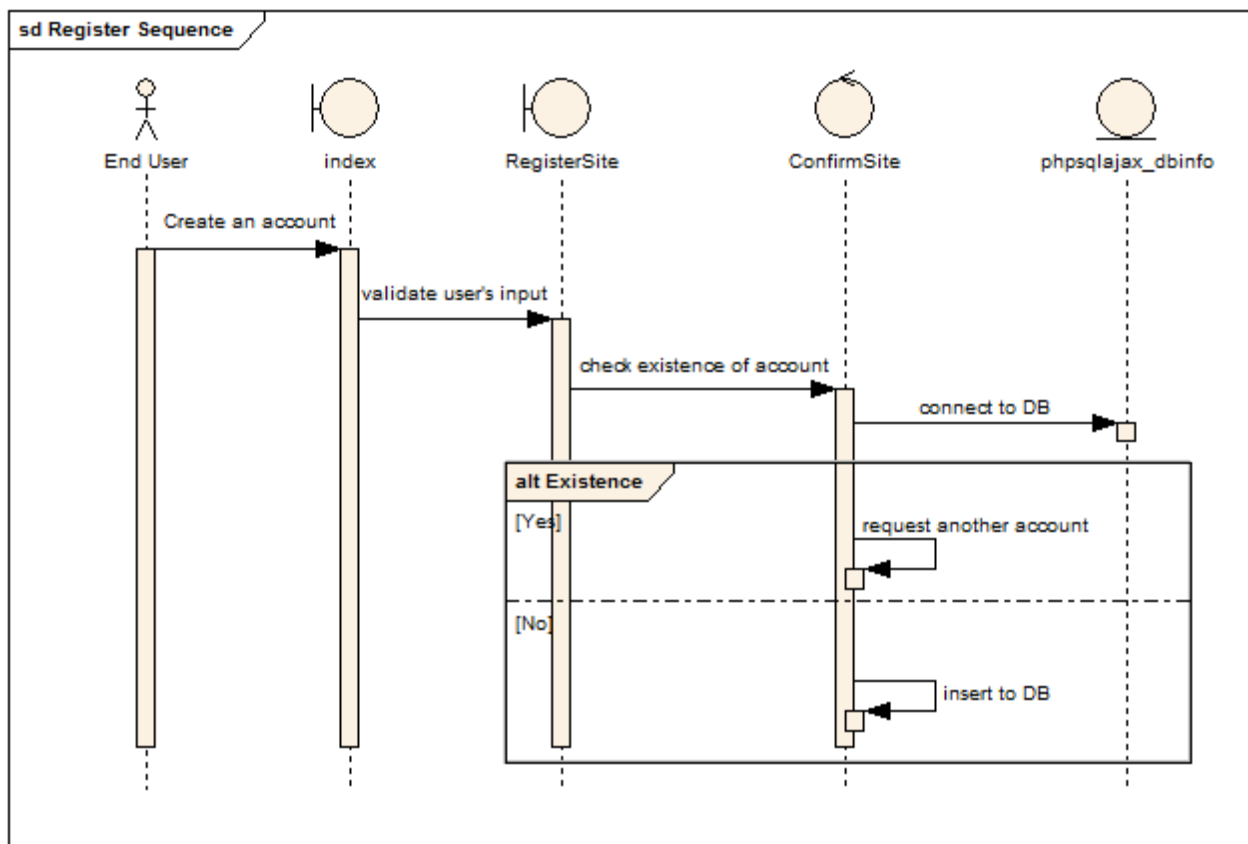


Figure 49 - Register Sequence Diagram

No	Method Name	Description
----	-------------	-------------

01	Create an account	-Description: Create new account when hasn't an account yet. -Parameter: N/A -Return: N/A
02	Validate user's input	-Description: Check valid data. -Parameter: N/A -Return: N/A
03	Check existence of account	-Description: Check user name, imei & email of account. In case has not existed yet, insert into database else re-input. -Parameter: N/A -Return: ArrayList<TrackingBean>
04	Connect to database	-Description: Connect database -Parameter: fileName -Return: N/A

Table 102 - Register Sequence Description

4.3.2.4 Update

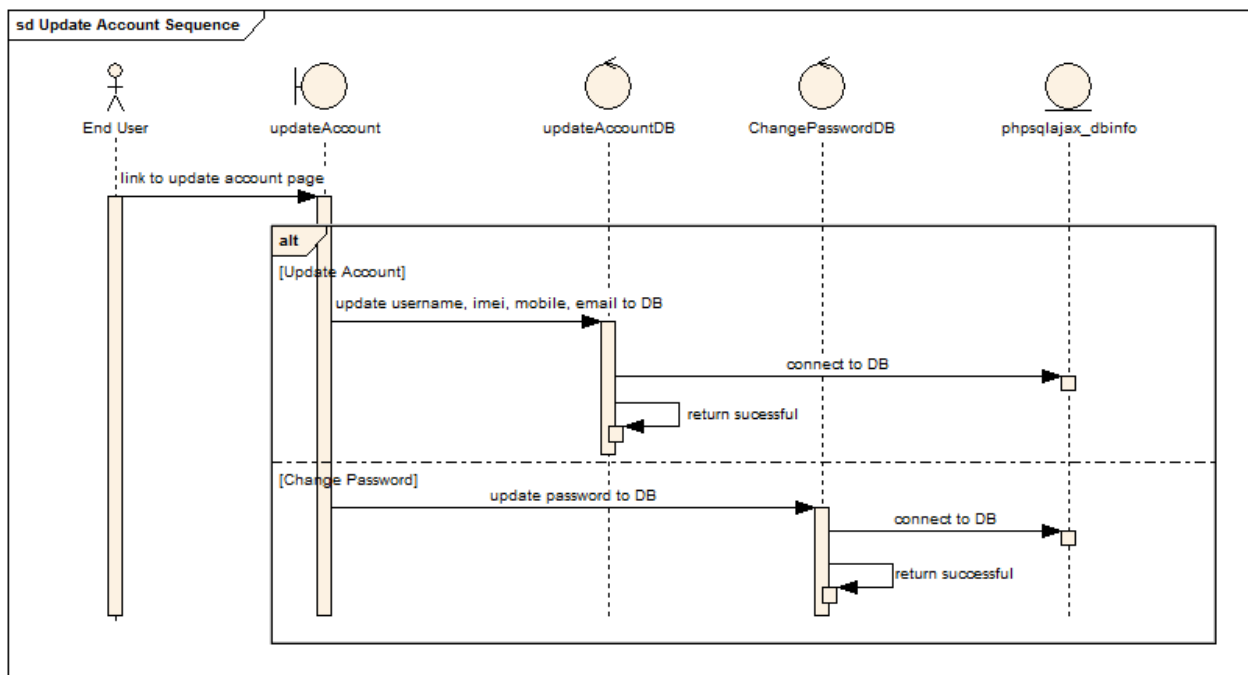


Figure 50 - Update Sequence Diagram

No	Method Name	Description
01	Link to update account page	-Description: Click on Update Account tab, go to this page. -Parameter: N/A -Return: N/A
02	Update account	-Description: Update information includes: user name,

	information	imei, email, mobile. -Parameter: N/A -Return: N/A
03	Connect to database	-Description: Connect database -Parameter: N/A -Return: N/A
04	Update password	-Description: If user changed password, update to database -Parameter: N/A -Return: N/A

Table 103 - Update Sequence Description

4.3.2.5 Show History

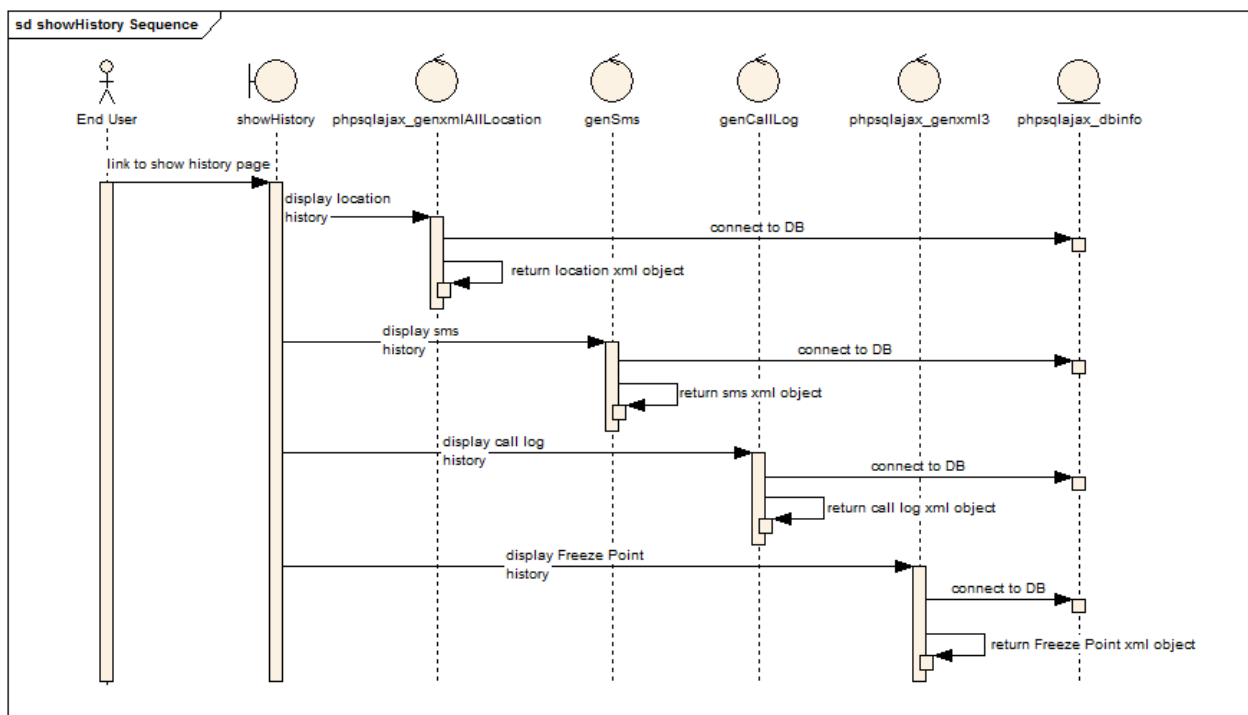


Figure 51 - Show History Sequence Diagram

No	Method Name	Description
01	Link to show history page	-Description: Ways to display history page: +After login successfully, go to this page. +Click on History tab on menu tab. -Parameter: N/A -Return: N/A
02	Display location history	-Description: Display position information on map. -Parameter: N/A -Return: N/A
03	Connect to database	-Description: Connect database

		-Parameter: N/A -Return: N/A
04	Display SMS history	-Description: Display SMS information. -Parameter: N/A -Return: N/A
05	Display call logs history	-Description: Display call logs information. -Parameter: N/A -Return: N/A
06	Display freeze point history	-Description: Display freeze point information on map. -Parameter: N/A -Return: N/A

Table 104 - Show History Sequence Description

4.3.2.6 Download/Upload

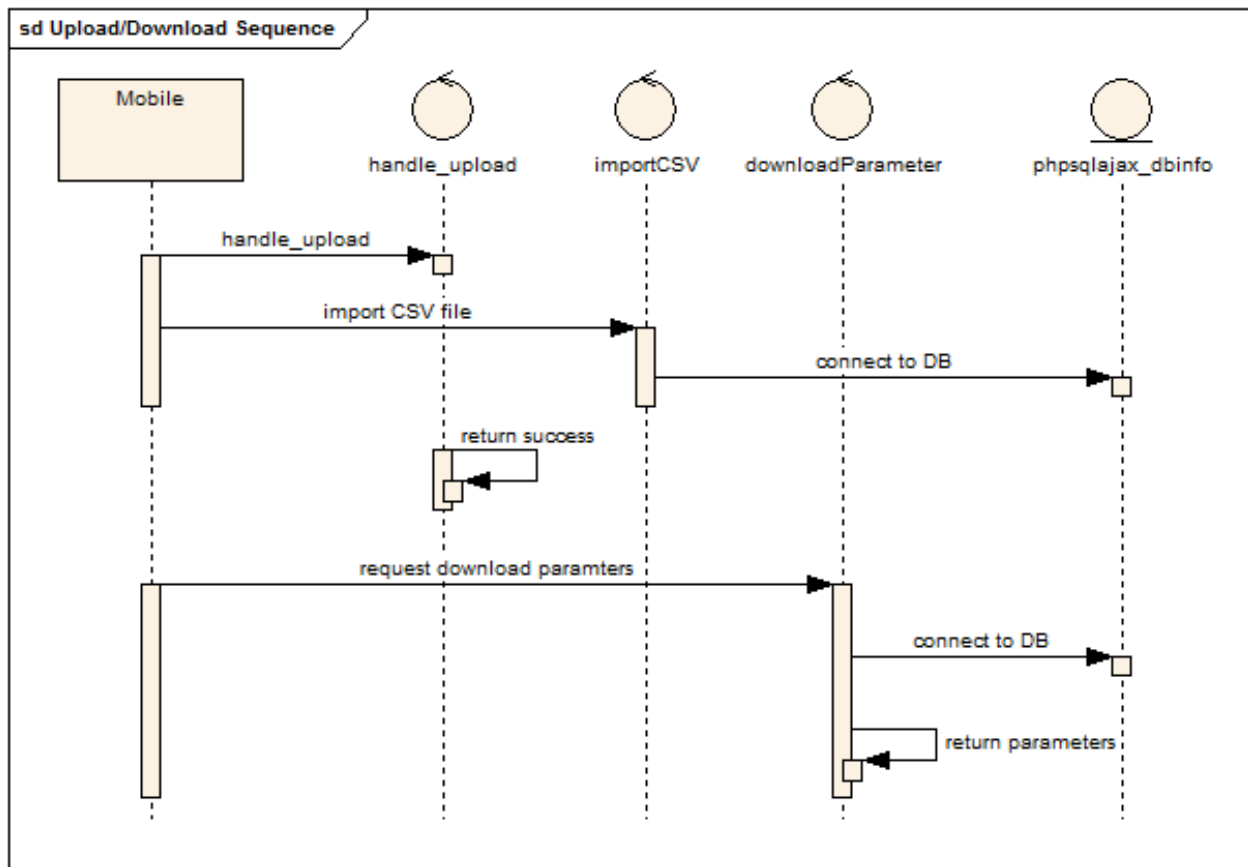


Figure 52 - Download/Upload Sequence Diagram

No	Method Name	Description
01	Handle_upload	-Description: After upload file from mobile to server, mobile send JSON object to server to inform having new upload file -Parameter: N/A -Return: N/A

02	Import CSV file	-Description: Read CSV file that upload from mobile to save to database. -Parameter: N/A -Return: N/A
03	Connect to database	-Description: Connect database -Parameter: N/A -Return: N/A
04	Request download parameter	-Description: When received request from mobile, server select parameter data from database. -Parameter: N/A -Return: N/A

Table 105 - Download/Update Sequence Description

4.4 Database

4.4.1 Mobile

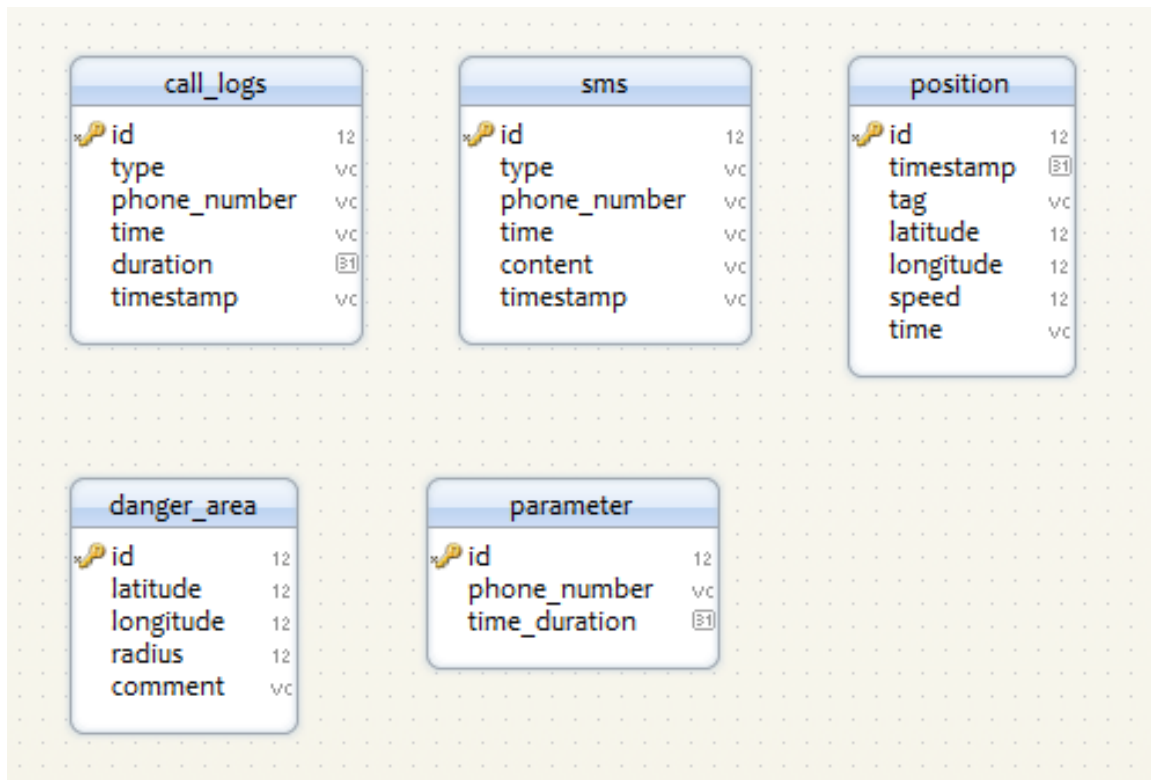


Figure 53 - Mobile Database

Table Name	No	Field Name	Data Type	Description
call_logs	01	id	int auto increment	Primary key. This is automatically increase value 1.
	02	type	varchar	Type of call.
	03	phone_number	text	Phone number of conversation.
	04	duration	text	Duration of conversation.
	05	time	time	Start time of call
	06	timestamp	text	Start timestamp of call.
sms	01	id	int auto increment	Primary key. This is automatically increase value 1.
	02	type	varchar	Type of sms
	03	phone_number	text	Phone number of conversation
	04	content	text	Content of sms
	05	time	time	Start time of sms
	06	timestamp	text	Start timestamp of sms
position	01	id	int auto increment	Primary key. This is automatically increase value 1.
	02	timestamp	text	Being timestamp position
	03	tag	varchar	Standard network to get position
	04	longitude	double	Longitude of position
	05	latitude	double	Latitude of position
	06	speed	double	Speed of moving
	07	time	text	Being time position
danger_area	01	id	int auto increment	Primary key. This is automatically increase value 1.
	02	latitude	double	Latitude of danger area
	03	longitude	double	Longitude of danger area
	04	radius	double	Radius of danger area
	05	comment	text	Comment of danger area
parameter	01	id	int auto increment	Primary key. This is automatically increase value 1.
	02	phone_number	text	Phone number of supervisor
	03	time_duration	time	Time to update data

Table 106 - Mobile Database Description

4.4.2 Server

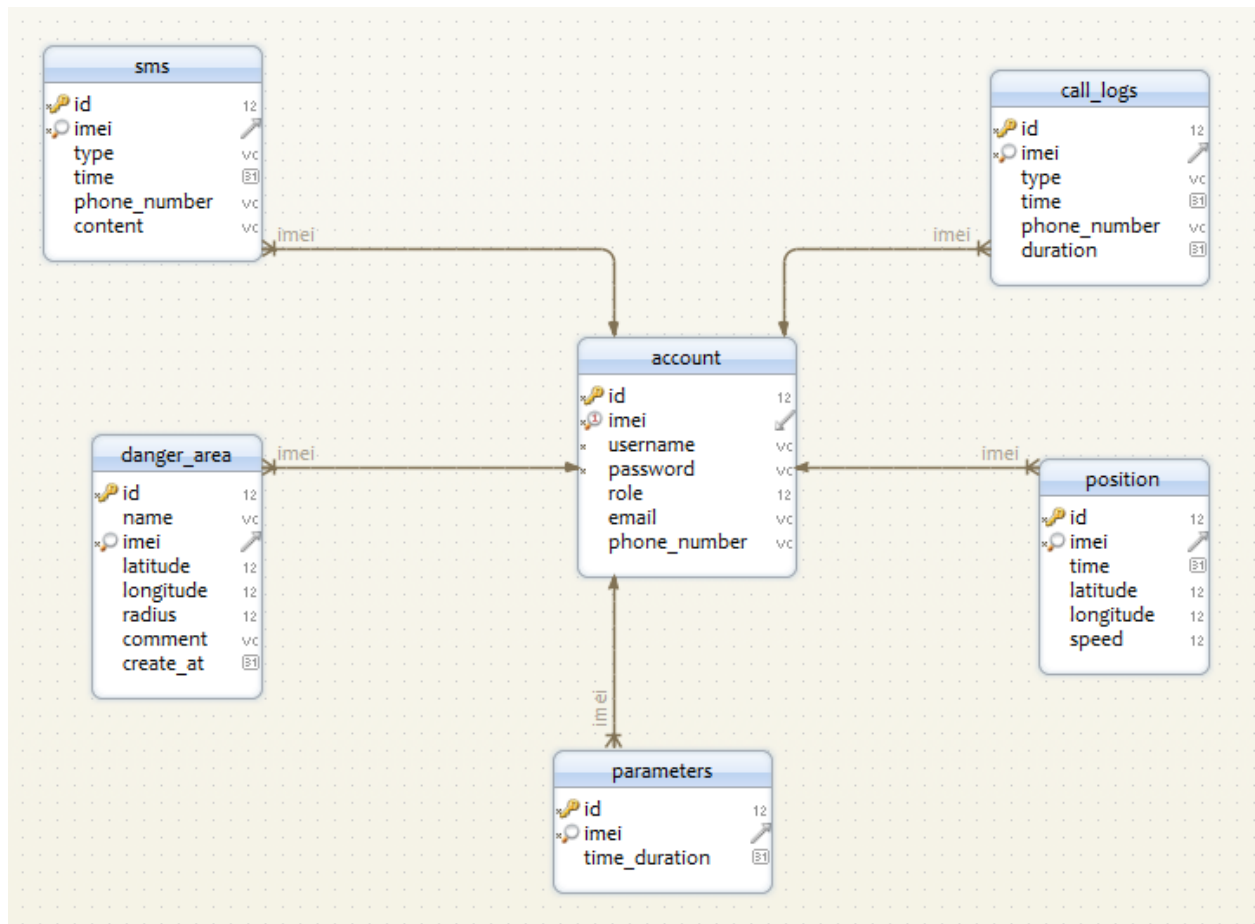


Figure 54 - Server Database

Table Name	No	Field Name	Data Type	Description
call_logs	01	id	int auto increment	Primary key. This is automatically increase value 1.
	02	imei	varchar	Imei of tracked mobile.
	03	type	varchar	Type of call.
	04	phone_number	text	Phone number of conversation.
	05	duration	text	Duration of conversation.
	06	time	time	Start time of call
	07	timestamp	text	Start timestamp of call.
sms	01	id	int auto increment	Primary key. This is automatically increase value 1.
	02	type	varchar	Type of sms
	03	phone_number	text	Phone number of conversation
	04	content	text	Content of sms
	05	time	time	Start time of sms
	06	timestamp	text	Start timestamp of sms.

	07	imei	varchar	Imei of tracked mobile.
position	01	id	int auto increment	Primary key. This is automatically increase value 1.
	02	timestamp	text	Being timestamp position
	03	tag	varchar	Standard network to get position
	04	longitude	double	Longitude of position
	05	latitude	double	Latitude of position
	06	speed	double	Speed of moving
	07	time	text	Being time position
	08	imei	varchar	Imei of tracked mobile
danger_area	01	id	int auto increment	Primary key. This is automatically increase value 1.
	02	name	String	Name of danger area
	03	latitude	double	Latitude of danger area
	04	longitude	double	Longitude of danger area
	05	radius	double	Radius of danger area
	06	comment	text	Comment of danger area
	07	imei	varchar	Imei of tracked mobile
	08	create_at	time	Created time
parameter	01	id	int auto increment	Primary key. This is automatically increase value 1.
	02	phone_number	text	Phone number of supervisor
	03	time_duration	time	Time to update data
	04	imei	varchar	Imei of tracked mobile
account	01	id	int auto increment	Primary key. This is automatically increase value 1.
	02	imei	varchar	Imei of tracked mobile
	03	username	varchar	Username of end users
	04	password	varchar	Password of end users
	05	roll	int	Permission
	06	email	text	Email of end users
	07	phone_number	text	Phone number of end users

Table 107 - Server Database Description

V. Chapter V - Testing

5.1 Requirements for Test

5.1.1 Feature to be tested

5.1.1.1 Application

- Setup Application on device.
- Application runs with normal conditions ex: turn on 3G, GPS
- Application runs with conditions ex: turn off 3G, GPS

- Action after device is received active message.
- Action when device stay at dangerous areas: Send message to number phone.
- Send information to server after customized time.
- Update new data from server
- Get information automatically and implicitly from device: GPS, Call Logs, SMS

5.1.1.2 Website

- UI of all Screens on Chrome and Firefox.
 - Login
 - Register
 - Profile
 - Danger
 - History
 - Admin
- User access and security
 - Check login, move to other screen, (on an account)
- Admin Screen.
 - Check user management: update (name, password, role), Delete account.
 - Display information correctly from data.
 - Check validation of all fields.
- History Screen
 - Show information of SMS, Call Logs, and Position correctly with time,
 - + Call logs: phone numbers, type, time, duration.
 - + SMS logs: Content, phone numbers, type, time.
 - Test all hyperlinks, zoom on map.
 - Show positions on map: name, time duration,
 - Switch between SMS, Call log tab.
 - Change Date dropdown list.
- Danger Area Screen.
 - Check create a new marker point
 - Check show marker point

- Check show marker point list
 - Check updates information of marker point.
 - Test all hyperlinks, zoom on map.
- Profile Screen.
 - Check validation of all fields.
 - Check updates information: IMEI, number phone, email, time.
 - Check buttons: update, cancel.
- Register Screen.
 - Check validation of all fields.
 - Button register.
- Login Screen.
 - Check validation of all fields.
 - Test all hyperlink: create an account, forgot password
 - Button sign in.
- Test on multiple browsers.
- Performance test:
 - Page loading on each click.
 - How many people can access at the same time...

5.1.2 Feature not to be tested

- Stable of application after device received active message. (3G and GPS are turn off)
- Try some special ways to remove application on device. (just normal)

5.2 Test Strategy

5.2.1 Test Types

5.2.1.1 Function Testing

Function testing of the target-of-test should focus on any requirements for test that can be traced directly to use cases or business functions and business rules. The goals of

these tests are to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules. This type of testing is based upon black box technique that is verifying the application and its internal processes by interacting with the application via the Graphical User Interface (GUI) and analyzing the output or results.

Table identified below is an outline of the testing recommended for each application.

Test Objective	Ensure proper target-of-test functionality, including navigation, data entry, processing, and retrieval.
Technique	Execute each use case, use-case flow, or function, using valid and invalid data, to verify the following: <ul style="list-style-type: none"> ❖ The expected results occur when valid data is used. ❖ The appropriate error or warning messages are displayed when invalid data is used. ❖ Each business rule is properly applied.
Completion Criteria	<ul style="list-style-type: none"> ❖ All planned tests have been executed. ❖ All identified defects have been addressed and closed
Special Considerations	Identify or describe those items or issues (internal or external) that impact the implementation and execution of function test

Table 108 - Outline Testing Recommended

5.2.1.2 User Interface Testing

User Interface (UI) testing verifies a user's interaction with the software. The goal of UI testing is to ensure that the User Interface provides the user with the appropriate access and navigation through the functions of the target-of-test. In addition, UI testing ensures that the objects within the UI function as expected and conform to corporate or industry standards.

Test Objective	<p>Verify the following:</p> <p>Navigation through the target-of-test properly reflects business functions and requirements, including window-to-window, field-to-field, and use of access methods (tab keys, mouse movements, accelerator keys)</p> <p>Window objects and characteristics, such as menus, size, position, state, and focus conform to standards.</p>
-----------------------	---

Technique	Create or modify tests for each window to verify proper navigation and object states for each application window and objects.
Completion Criteria	Each window successfully verified to remain consistent with benchmark version or within acceptable standard
Special Considerations	Not all properties for custom and third party objects can be accessed.

Table 109 - User Interface Testing

5.2.1.3 Performance Testing

Performance profiling is a performance test in which response times, transaction rates, and other time-sensitive requirements are measured and evaluated. The goal of Performance Profiling is to verify performance requirements have been achieved. Performance profiling is implemented and executed to profile and tune a target-of-test's performance behaviors as a function of conditions such as workload or hardware configurations.

Test Objective	Verify performance behaviors for designated transactions or business functions under the following conditions: <ul style="list-style-type: none"> ❖ normal anticipated workload ❖ anticipated worst case workload
Technique	<ul style="list-style-type: none"> ❖ Use Test Procedures developed for Function or Business Cycle Testing. ❖ Modify data files to increase the number of transactions or the scripts to increase the number of iterations each transaction occurs. ❖ Scripts should be run on one machine (best case to benchmark single user, single transaction) and be repeated with multiple clients (virtual or actual, see Special Considerations below).
Completion Criteria	<ul style="list-style-type: none"> ❖ Single Transaction or single user: Successful completion of the test scripts without any failures and within the expected or required time allocation per transaction. ❖ Multiple transactions or multiple users: Successful completion of the test scripts without any failures and within acceptable time allocation.
Special Considerations	Comprehensive performance testing includes having a background workload on the server. There are several methods that can be used to perform this, including:

	<ul style="list-style-type: none"> ❖ “Drive transactions” directly to the server, usually in the form of Structured Query Language (SQL) calls. ❖ Create “virtual” user load to simulate many clients, usually several hundred. Remote Terminal Emulation tools are used to accomplish this load. This technique can also be used to load the network with “traffic”. ❖ Use multiple physical clients, each running test scripts to place a load on the system. <p>Performance testing should be performed on a dedicated machine or at a dedicated time. This permits full control and accurate measurement.</p> <p>The databases used for Performance Testing should be either actual size or scaled equally.</p>
--	--

Table 110 - Performance Testing

5.2.2 Test Environment

Test client configuration:

- a. Hardware
 - Samsung galaxy S2
 - Intel core i3 2310M 2.1Ghz
- b. Software
 - Microsoft SQL 2008 R2
 - QTrace
 - Jmeter or Open STA
 - Chrome 23 and Firefox 16
 - Firebug
 - Simulator of android

5.3 Test cases

5.3.1 Application

5.3.1.1 Set Up

Module Code		Application Set Up		
ID	Test Case Description	Test Case Procedure	Expected Output	Result

A1	Display	1. Root Android device. (Login by Admin account.) 2. Set up application. 3. Unroot Android device and Login by normal account.	1. There is no icon of app on all screens. 2. There is no app on setting menu.	Pass
A2	Confirm after setup	1. Root Android device. (Login by Admin account.) 2. Set up application. 3. Login by normal account. 4. Turn on 3G and GPS 5. Access to web for checking.	1. Device will send to serve a confirmation include: + IMEI of device. + Current location.	Untested
A3	Check unmemorable	1. Root Android device. (Login by Admin account.) 2. Set up application. 3. Login by normal account. 4. Find some normal ways to remove this application ex: +find icon and remove +find it on setting menu...	1. There is not a normal way to remove this application.	Untested
A4	Check automatically after reset device.	1. After set up successfully on device then reset device. 2.Go to setting, App, manage applications, running.	1. You can see it running with 3 processes.	Pass

Table 111 - Set Up

5.3.1.2 Running with 3G &GPS Are Turned On

Module Code		Application - 3G & GPS Are Turned On		
ID	Test Case Description	Test Case Procedure	Expected Output	Result
A5	Get GPS information and save correctly in database. With Stationary.	1. Setup and turn on 3G and GPS. 2. Stationary. 3. Check data have just saved by using it on GOOGLE. 4. Check data save correctly.	1. Data what have just saved is right. 2. Data saving correctly for example: right columns, rows, table, file	Pass
A6	Get GPS information and save correctly in database. With Move.	1. Setup and turn on 3G and GPS. 2. Move. 3. Check data have just saved by using it on GOOGLE. 4. Check data save correctly.	1. Data what have just saved is right. 2. Data saving correctly for example: right columns, rows, table, file...	Pass

A7	Send GPS data to server after customize time. With Stationary.	1. Setup and turn on 3G and GPS. 2. Stationary. 3. Set configure time on web site (1 or 2 minutes). 4. Check data on server database and compare them with data on device	1. After customize time Data will be sent to server. 2.Data appear correctly on server database	Pass
A8	Get message information and save in database.	1. Setup and turn on 3G. 2. Send a message to another number. 3. Send a message from another number to this device. 3. Check database on device.	1. Data will be saved on database on device after send or receive messages. 2. All sending and receiving numbers are right.	Pass
A9	Send message data to server after customize time.	1. Setup and turn on 3G and GPS. 2. Set configure time on web site (1 or 2 minutes). 3. Check data on server database and compare them with data on device.	1. After customize time Data will be sent to server. 2. Data appear correctly on server database. + All sending and receiving numbers are right.	Pass
A10	Get Call log information and save in database.	1. Setup and turn on 3G. 2. Call to another number. 3. Call from another number to this number of device. 3. Check database on device.	1. Data will be saved on database on device after send or receive messages. 2. Going, Coming and missing call have saved on database of device.	Pass

A11	Send call log data to server after customize time.	1. Setup and turn on 3G and GPS. 2. Set configuration time on web site (1 or 2 minutes). 3. Check data on server database and compare them with data on device.	1. After customize time Data will be sent to server. 2. Data appear correctly on server database. + Going, Coming and missing call have saved on database of server.	Pass
-----	--	---	--	------

Table 112 - 3G & GPS Are Turned On

5.3.1.3 Check Danger Areas

Module Code		Application - Check Danger Areas		
ID	Test Case Description	Test Case Procedure	Expected Output	Result
A12	Check when device doesn't go to dangerous area.	1. Setup and turn on 3G and GPS on device. 2. Set up dangerous point, time, distance... on website. 3. Stand at location outside this area.	1. There is no alert on number phone what have registered.	Pass
A13	Check when go to dangerous area	1. Setup and turn on 3G and GPS on device. 1. Set up dangerous point, distance... on website. 2. Check data on database of device. 3. Stand at location inside this area with time less customized time on website.	1. There is no alert on number phone what have registered.	Pass

A14	Check when go to 2 different dangerous area	1. Setup and turn on 3G and GPS on device. 2. Set up some of dangerous point, time, distance... on website. 3. Check data on database of device. 4. Stand at location inside this area with customized time on website. 5. Go and stand at another dangerous area with customized time on website for this area.	1. There is an alert on number phone what have registered. 2. There is another alert to number phone what have registered.	Pass
A15	Check when go one dangerous area twice or more	1. Setup and turn on 3G and GPS on device. 2. Set up some of dangerous point, time, distance... on website. 3. Check data on database of device. 4. Stand at location inside this area with customized time on website. 5. Go outside and comeback then standing at this dangerous area with customized time on website for this area.	1. There is an alert on number phone what have registered. 2. There is another alert to number phone what have registered.	

Table 113 - Check Danger Areas

5.3.1.4 Check Receive Active Message

Module Code		Application - Receive Active Message		
ID	Test Case Description	Test Case Procedure	Expected Output	Result

A16	Check activation after device receives message.	1. Setup and turn off 3G and GPS on device. 2. Set up dangerous point, time, distance... on website. 3. Send Active message to device following form: ON 4. Check turn on GPS + 3G. 5. Check turn on update information.	1. 3G and GPS are turned on Auto. 2. Data is update on database of device.	Fail
A17	Check delete message after device receives message.	1. Setup and turn off 3G and GPS on device. 2. Send Active message to device following form: ON 3. Check message on inbox.	1. Message is deleted and 3G and GPS are turned on Auto.	Pass

Table 114 - Receive Active Message

5.3.2 Central Web with Chrome

5.3.2.1 Login Screen

Module Code		Website on Chrome – Login Screen		
ID	Test Case Description	Test Case Procedure	Expected Output	Result
C1	Check Password textbox display	1. Go to Web site http://giangvh007.zxq.net/ 2. Enter some characters on password textbox for ex: tungbach...	Password textbox will show on dot type.	Pass
C2	Check Create an Account (hyperlink)	1. Go to Web site http://giangvh007.zxq.net/ 2. Click on Create an account.	Going on Register Screen.	Pass
C3	Check Forget password (hyperlink)	1. Go to Web site http://giangvh007.zxq.net/ 2. Click on forget password	Going on forget password Screen.	Fail
C4	Check Sign in button with right data	1. Go to Web site http://giangvh007.zxq.net/ 2. Enter right User and password. 3. Click Sign In button.	Going on History Screen.	Pass

C5	Check Sign in button with wrong data	1. Go to Web site http://giangvh007.zxq.net/ 2.Enter right User and wrong password. 3. Click Sign In button.	Display a error message See : login-fail.jpg	Pass
C6	Check Sign in button with wrong data	1. Go to Web site http://giangvh007.zxq.net/ 2.Enter wrong User 3.Click Sign In button.	Display a error message See : login-fail.jpg	Pass
C7	Check Sign in button with nothing input	1. Go to Web site http://giangvh007.zxq.net/ 2.Click Sign In button.	Display a error message See : login-fail.jpg	Pass

Table 115 - Website on Chrome - Login Screen

5.3.2.2 Register Screen

Module Code		Website on Chrome - Register Screen		
ID	Test Case Description	Test Case Procedure	Expected Output	Result
C8	Check min length of User textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on Create an account link 2.Enter 5 characters or less to User textbox example :12345	Display an error message please see image. Error-register.jpg	Pass
C9	Check max length of User textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on Create an account link 2.Try enter 13 characters or more to User textbox example 12345567890123	Just can input max 12 characters	Pass

C10	Check enter special characters User textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on Create an account link 2.Try enter some special characters to User textbox (by normal way)	You cannot enter them. You just can enter number and normal characters. For ex: #####bach123	Pass
C11	Check min length of Password textbox	1.Go to Web site http://giangvh007.zxq.net/ 2.Enter 5 characters to Password textbox example 12345	Display an error message please see image. Error-register.jpg	Pass
C12	Check max length of Password textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on Create an account link 2.Try enter 13 characters or more Password textbox example 12345567890123	Just can input max 12 characters	Pass
C13	Check enter special characters of Password textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on Create an account link 2.Try enter some special characters to Password textbox example :\$%^&*(by normal way)	You cannot enter them. You just can enter number and normal characters. For ex @\$\$@#4bach123	Pass
C14	Check min length of Confirm Password textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on Create an account link 2.Enter 5 characters to confirm password textbox example 12345	Display an error message please see image. Error-register.jpg	Pass

C15	Check max length of Confirm Password textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on Create an account link 2.Try enter 13 characters or more to confirm password textbox example 12345567890123	Just can input max 12 characters	Pass
C16	Check enter special characters of confirm Password textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on Create an account link 2.Try enter some special characters to confirm password textbox \$%^&*(by normal way)	You cannot enter them. You just can enter number and normal characters. For ex:bach123	Pass
C17	Confirm Password textbox	1. Go to Web site http://giangvh007.zxq.net/ and click on Create an account link 2. Enter a password to password textbox 3. enter a different password to above password to confirm password to confirm password textbox	Display an error message See : error-register.jpg	Pass
C18	Check IMEI text box with enter 13 characters or less	1.Go to Web site http://giangvh007.zxq.net/ 2.Try enter 13 characters or less example 1234567890123	Display an error message See : error-register.jpg	Pass
C19	Check IMEI text box with enter 16 characters or more	1.Go to Web site http://giangvh007.zxq.net/ and click on Create an account link 2.Try enter 16 characters or more example 1234567890123456	You cannot enter 16 characters.	Pass
C20	Check IMEI text box with enter 15 characters or more	1.Go to Web site http://giangvh007.zxq.net/ and click on Create an account link 2.Try enter 15 characters example 123456789012345	There is no message.	Pass

C21	check enter characters (tungbach) to IMEI textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on Create an account link 2.Try enter some characters to IMEI textbox fx :tungbach123	you just can input number	Pass
C22	check enter characters to Mobile textbox	1.Go to Web site http://giangvh007.zxq.net/and click on Create an account link 2.Try enter character to Mobile textbox fx:bach123	you just can input number	Pass
C23	Check form for Email textbox	1.Go to Web site http://giangvh007.zxq.net/and click on Create an account link 2.Try enter wrong form of an email fx: bach99@ @ @yaho.om	Display an error message See : error-register.jpg	Pass
C24	Check Register with right data	1. Go to Web site http://giangvh007.zxq.net/ 2.Click on Create an account. 3. Fill right any fills. 4. Click Register button.	1. Data will be saved in database.	Pass
C25	Check Register with wrong data	1. Go to Web site http://giangvh007.zxq.net/ 2.Click on Create an account. 3. Fill wrong one fill.(user username) 4.Click Register button.	1. Display a error message. See :error-register.jpg	Pass
C26	Check Summit button with nothing	1. Go to Web site http://giangvh007.zxq.net/and click on Create an account link 2.Click on Create an account. 3. Click Summit button.	1. Display a error message. See : summit-error-register.jpg	Pass
C27	Check click login active.	1. Go to Web site http://giangvh007.zxq.net/and click on Create an account link 2.Click on Create an account. 3. Click Login button.	Going on Login Screen.	Pass
C28	Check username exist	1. Go to Web site http://giangvh007.zxq.net/ 2.Click on Create an account. 3. Enter a username what exited on database. for example, bachpt 4. Fill remain fill with right data 5. click on Summit button	Display a error message See :register-error name.jpg	Pass

C29	Check IMEI exist	1. Go to Web site http://giangvh007.zxq.net/ 2. Click on Create an account. 3. Enter an IMEI what existed on database. for example 000000000000,359199020254432 4. Fill remain fill with right data 5. click on Summit button	Display a error message see iemi-exist-register.jpg	Pass
C30	Check Email exist	1. Go to Web site http://giangvh007.zxq.net/ 2. Click on Create an account. 3. Enter a Email what existed on database. for example gbach90@yahoo.com 4. Fill remain fill with right data 5. click on Summit button	Display a error message, see email-exist-register.jpg	Pass
C31	Check mobile exist	1. Go to Web site http://giangvh007.zxq.net/ 2. Click on Create an account. 3. Enter a Email what existed on database. for example tunbach90@yahoo.com 4. Fill remain fill with right data 5. click on Summit button	Display a error message ,see phone-exist-register.jpg	Pass

Table 116 - Website on Chrome - Register Screen

5.3.2.3 Update Account Screen

Module Code		Website on Chrome – Update Account Screen		
ID	Test Case Description	Test Case Procedure	Expected Output	Result
C32	User name field	1. Go to Web site http://giangvh007.zxq.net/ and click on update account tab 2. Try to enter some characters to username field	Cannot enter anything to username field	Pass
C33	IMEI text box	1. Go to Web site http://giangvh007.zxq.net/and click on update account tab 2. Try enter 13 characters or less to IMEI text box example 1234567890123	Display an error message see error-update1.jpg	Pass

C34	IMEI text box	1.Go to Web site http://giangvh007.zxq.net/and click on update account tab 2.Try enter 16 characters or less to IMEI text box example 1234567890123456	You cannot enter 16 characters.	Pass
C35	IMEI text box	1.Go to Web site http://giangvh007.zxq.net/and click on update account tab 2.Try enter 15 characters or less to IMEI textbox example 123456789012345	There is no message.	Pass
C36	Mobile textbox	1.Go to Web site http://giangvh007.zxq.net/and click on update account tab 2.Try enter character to mobile textbox fxtungbach123	You just can input number	Pass
C37	Email textbox	1.Go to Web site http://giangvh007.zxq.net/and click on update account tab 2.Try enter wrong form of an email text box fx tungbach99@ @.yaho.com	Display an error message see error- update1.jpg	Pass
C38	Check Update button with nothing change	1.Go to Web site http://giangvh007.zxq.net/ and click on update account tab 2.Login success 3.Click on update account tab 4.Click update button	1. Login success 2.Display profile screen 3.There is nothing change in database	Pass
C39	Check Update button with change	1.Go to Web site http://giangvh007.zxq.net/and click on update account tab 2.Login success 3.Click on Profile 4.Change phone number or email ... 5.Click update button	1. Login success 2.Display profile screen 3.Data is updated in database.	Pass
C40	Check History Tab	1.Go to Web site http://giangvh007.zxq.net/and click on update account tab 2.Login success 3.Click on Profile 4.Click on History tab	Going on History screen.	Pass

C41	Check Danger Area Tab	1.Go to Web site http://giangvh007.zxq.net/ and click on update account tab 2.Login success 3.Click on Profile 4.Click on Danger Area hyperlink	Going on Danger Area screen.	Pass
C42	Check Log out	1.Go to Web site http://giangvh007.zxq.net/ 2.Login success and click on update account tab 3.Click on Profile 4.Click on Log out hyperlink	Back to login screen	Pass
C43	Check radio button Delete Data Tracking	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.Click on update account tab 4.tick on Delete Data Tracking 5.chose a day what has data of GPS by click on calendar icon. 6.Click on update button	1. Display a success message or success page 2. Data will be deleted on database.	Pass
C44	Check radio button Delete Data SMS	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.Click on update account tab 4.tick on Delete Data SMS 5.chose a day what has data of GPS by click on calendar icon. 6.Click on update button	1. Display a success message or success page 2. Data will be deleted on database.	Pass
C45	Check radio button Delete Data Call Log	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.Click on update account tab 4.tick on Delete Data Call Log 5.chose a day what has data of GPS by click on calendar icon. 6.Click on update button	1. Display a success message or success page 2. Data will be deleted on database.	Pass
C46	Check all radio button	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.Click on update account tab 4.tick on Delete Data Call Log, Delete Data SMS, Delete Data Tracking 5.chose a day what has data of GPS by click on calendar icon. 6.Click on update button	1. Display a success message or success page 2. Data will be deleted on database.	Pass

C47	Check min length of old Password textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on update account tab 2.Enter 5 characters to old password textbox example 12345	Display an error message please see image. error-update2.jpg	Pass
C48	Check enter special characters of old Password textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on update account tab 2.Try enter some special characters to old password textbox /\$%^&*(by normal way)	You cannot enter them. You just can enter number and normal characters. For ex :tungbach123	Pass
C49	Check min length of New Password textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on update account tab 2.Enter 5 characters to new password textbox example 12345	Display an error message please see image. error-update2.jpg	Pass
C50	Check max length of New Password textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on update account tab 2.Try enter 13 characters or more to new password textbox example 12345567890123	Just can input max 12 characters	Pass
C51	Check enter special characters of old Password textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on update account tab 2.Try enter some special characters to old password text box :\$%^&*(by normal way)	You cannot enter them. You just can enter number and normal characters. For ex tungbach123	Pass
C52	Check min length of Confirm Password textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on update account tab 2.Enter 5 characters to confirm password example 12345	Display an error message please see image. error-update2.jpg	Pass

C53	Check max length of Confirm Password textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on update account tab 2.Try enter 13 characters or more to confirm password :example 12345567890123	Just can input max 12 characters	Pass
C54	Check enter special characters of confirm Password textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on update account tab 2.Try enter some special characters to confirm password : \$%^&*(by normal way)	You cannot enter them. You just can enter number and normal characters. For ex tungbach123	Pass
C55	Confirm Password textbox	1.Go to Web site http://giangvh007.zxq.net/ and click on update account tab 2. Enter a password different to password in password field	Display an error message error-update2.jpg	Pass
C56	Check button change password	1. Go to Web site http://giangvh007.zxq.net/ and click on update account tab 2. Fill 3 fields: old password, new password, confirm password with right data 4. click button change password	Display a success page	Pass
C57	Check button change password with wrong data	1. Go to Web site http://giangvh007.zxq.net/ and click on update account tab 2. Fill 3 fields: old password, new password, confirm password with wrong data 4. click button change password	Display error page	Pass
C58	Check display in 3 password fields old password, new password, confirm password	1. Go to Web site http://giangvh007.zxq.net/ and click on update account tab 2. Fill 3 field old password, new password, confirm password	Display dot type	Pass

C59	Check IMEI exist	1. Go to Web site http://giangvh007.zxq.net/ 2.Going on update account page 3.Enter an IMEI what exited on database. for example 000000000000,359199020254432 4. Fill remain fill with right data 5. click on Update button	Display a error message error-register.jpg	Pass
C60	Check Email exist	1. Go to Web site http://giangvh007.zxq.net/ 2.Going on update account page 3.Enter a Email what exited on database. for example tungbach90@yahoo.com 4. Fill remain fill with right data 5. click on Update button	Display a error message error-register.jpg	Pass
C61	Check mobile exist	1. Go to Web site http://giangvh007.zxq.net/ 2.Going on update account page 3.Enter a Email what exited on database. for example tungbach90@yahoo.com 4. Fill remain fill with right data 5. click on Update button	Display a error message error-register.jpg	Pass

Table 117 - Website on Chrome - Update Account Screen

5.3.2.4 Danger Area Screen

Module Code		Website on Chrome – Danger Area Screen		
ID	Test Case Description	Test Case Procedure	Expected Output	Result
C62	Time duration	1.Go to Web site http://giangvh007.zxq.net/ and click on danger area tab 2.Try enter character fx: tungbach123 on Time upload data field	You just can input number	Pass
C63	Check ok button	1. Go to Web site http://giangvh007.zxq.net/ 2. Click on danger tab 3. Enter a number to Time upload data field, for ex 10000 click ok	A success message box will be showed see: parameter-update.jpg	Fail

C64	Check display default with data about danger areas.	1.Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.Click on Danger Area tab	1. Screen look like following image 2. Data is displayed right data in database name ,Radius , comment, +Number of Marker +Position of Marker	Pass
C65	Check creates a new danger point.	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success. 3. Click on Danger Area tab 4. click on a point on map, 5 fill right data on 3 fields 6 click save button	1. After save data will be save on database. 2,Show this point on map.	Pass
C66	Check min length of Name textbox	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success. 3. Click on Danger Area tab 4. click on a point on map, 5 enter 5 characters or less to name textbox 6 click save button	Display an error message	Pass
C67	Radius textbox	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success. 3. Click on Danger Area tab 4. click on a point on map, 5 Try to enter some characters or less to radius textbox 6 click save button	You just can input number	Pass
C68	Check Close button	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.Click on Danger Area tab 4. click on a point on map, 5.Click on close button.	Disappear the box.	Pass

C69	Check click on marker name	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.Click on Danger Area hyperlink 4. click on a point on map, 5.Click random on marker name	This point will be showed. +Show position +Show box with name, radius, and comment note: see image	Pass
C70	Check List of danger area	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success 3. Click on danger tab 4.click on random point of list of danger area	1. All danger points will be showed. 2. Danger point will be showed. if there is no data in database, then SMS table will show "Khong co du lieu cho ngay "	Pass
C71	Check zoom in zoom out move right , left ,up and down.	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.Click on Danger Area hyperlink 4. Use zoom in , zoom out, move right , left ,up and down.	They work well.	Pass
C72	Check Save & close button	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.Click on Danger Area tab 4. Click on a marker point on map, 5.try to change name, radius, and comment. 6. click save & close button	1. Display success message box 2. Data what has changed will be saved to database. See: error-register.jpg	Fail
C73	Check Delete button	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.Click on Danger Area tab 4. Click on a marker point on map, 5.try to change name, radius, and comment. 6. click save & close button	1. Display success message box 2. data what has chosen will be delete on database see marker-delete.jpg	Pass

Table 118 - Website on Chrome - Danger Area Screen

5.3.2.5 History Screen

Module Code		Website on Chrome – History Screen		
ID	Test Case Description	Test Case Procedure	Expected Output	Result

C74	Check default of display	1.Go to Web site http://giangvh007.zxq.net/ 2.Login success	1. Screen will display all data of current day. +SMS log information +positions on map	Pass
C75	Check Day calendar icon	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.Click on calendar icon 4. choose random day	1. The day what has chosen will be display on textbox beside 2. All data of SMS and call log will be showed on 2 table (SMS, call logs)	Pass
C76	SMS table	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.Click on day calendar icon 4. choose random day	1.Screen will show exactly data in database - type, phone number, time, content... if there is no data in database, then SMS table will show "Khong co du lieu cho ngay "	Pass
C77	Call Logs table	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.Click on day calendar icon 4. choose random day	1.Screen will show exactly data in database - type, phone number, time, duration if there is no data in database, then SMS table will show "Khong co du lieu cho ngay "	Pass
C78	Check zoom in zoom out move right , left ,up and down.	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.Use zoom in, zoom out, move right , left ,up and down.	They work well.	Pass

C79	Check click History tab	1.Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.Click on History tab	Going on History screen.	Pass
C80	Check click Danger Area tab	1.Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.Click on Danger Area tab	Going on Danger Area screen.	Pass
C81	Check click Log out	1.Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.Click on Log out	Back to login screen	Pass
C82	Check List of location	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success 3.click on start or end.	1. Start and end point, distance will be showed, 2. After click on start or end point then start or end point will be showed on map. if there is no data in database, then SMS table will show "Khong co du lieu cho ngay "	Pass
C83	Check List of danger area	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success 3. click on random point of list of danger area	1. All danger points will be showed, 2. Danger point will be showed. if there is no data in database, then SMS table will show "Khong co du lieu cho ngay "	Pass
C84	Check Map display	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success 3. click on random point of list of danger area 4. Click on marker point on map	1. All danger points will be showed, 2. danger point will be showed. 3. After click on a marker point on map, information box will be showed.	Pass

Table 119 - Website on Chrome - History Screen

5.3.2.6 Admin Screen

Module Code		Website on Chrome – Admin Screen		
ID	Test Case Description	Test Case Procedure	Expected Output	Result
C85	Check data show correctly	1. Go to Web site http://giangvh007.zxq.net/ 2. Login success by Admin account	Data show correctly with database	Pass
C86	Check pop up after click on a row	1. Go to Web site http://giangvh007.zxq.net/ 2. Login success by Admin account 3. Click on a row.	Pop up will be showed see : pop-up.jpg	Pass
C87	Check ID and username field	1. Go to Web site http://giangvh007.zxq.net/ 2. Login success by Admin account Click on a row. 3. Try to change Id and username field	You cannot change id and username field	Pass
C88	Role dropdown list	1. Go to Web site http://giangvh007.zxq.net/ Login success by Admin account Click on a row. 2. Try to select one of three options	The option what has chosen will be showed see :roles.jpg	Pass
C89	Delete button	1. Go to Web site http://giangvh007.zxq.net/ 2. Login success by Admin account 3. Click on a row on table 4. click delete button	This row disappears and this data is deleted in database.	Pass
C90	Check IMEI exist	1. Go to Web site http://giangvh007.zxq.net/ 2. Login success by Admin account and click on a row 3. Enter a IMEI what existed on database. for example :000000000000,35919902025443 2 4. Fill remain fields with right data 5. click on Update button	Display a error message imei-exist-admin.jpg	Pass

C91	Check Email exist	1. Go to Web site http://giangvh007.zxq.net/ 2.Going on update account page 3.Enter a Email what exited on database. for example tunbach90@yahoo.com 4. Fill remain fields with right data 5. click on Update button	Display a error message imei-exist-admin.jpg	Pass
C92	Check mobile exist	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success by Admin account Click on a row. 3. Enter a Email what exited on database. for example tunbach90@yahoo.com 4. Fill remain fields with right data 5. click on Update button	Display a error message mobile-exist-admin.jpg	Pass
C93	IMEI text box	1. Go to Web site http://giangvh007.zxq.net/Login success by Admin account Click on a row. 2.Try enter 13 characters or less example 1234567890123	Display an error message	Pass
C94	IMEI text box	1. Go to Web site http://giangvh007.zxq.net/ with admin account. 2. Click on a row. 3. After pop up was showed 2.Try enter 14 characters or more example on IMEI box. 1234567890123 5. click update button	Display an error message	Fail

C95	IMEI text box	1. Go to Web site http://giangvh007.zxq.net/ Login success by Admin account Click on a row. 2.Try enter 16 characters or more example 1234567890123456	You cannot enter 16 characters.	Pass
C96	IMEI text box	1. Go to Web site http://giangvh007.zxq.net/ Login success by Admin account Click on a row. 2.Try enter 15 characters or morehttp://giangvh007.zxq.net/ example 123456789012345	There is no message.	Pass
C97	Mobile textbox	1. Go to Web site http://giangvh007.zxq.net/ Login success by Admin account Click on a row. 2.Try enter character fxtungbach123	You just can input number	Pass
C98	Email textbox	1.Go to Web site http://giangvh007.zxq.net/ Login success by Admin account Click on a row. 2.Try enter wrong form of an email to email textbox fx tungbach99@@@yaho.com	Display an error message see email-error- admin.jpg	Pass
C99	Check x (close) button of pop up.	1. Go to Web site http://giangvh007.zxq.net/ 2.Login success by Admin account 3.click on a row. 4. clean some field 5.click x button	1. pop up will be closed 2. nothing change on database	Pass

C100	Check Log out	1.Go to Web site http://giangvh007.zxq.net/ 2.Login success by Admin account 3.click on log out	Back to login screen	Pass
------	---------------	--	----------------------	------

Table 120 - Website on Chrome - Admin Screen

5.4 Test Summary Report

Test Execution Summary			
	Pass	Fail	Number of Test cases
Function	190	0	190
Permission	8	0	8
Performance	8	0	8
Test coverage			
Test coverage	206	0	100%
Test successful coverage			100%

Table 121 – Test Execution Summary

5.4.1 Test Execution Summary

5.4.2 Test Results

5.4.2.1 Summary of Incident Reports Raised During Project

Severity	Total Defect	Open Defect			
		Assigned	Corrected	Accepted	Total Open defects
Fatal	0	0	0	0	0
Serious	4	4	3	3	4
Medium	36	36	36	36	36
Cosmetic	0	0	0	0	4

Total (Status)	40	40	39	36	40
-----------------------	-----------	-----------	-----------	-----------	-----------

Table 122 - Summary of Incident Report

Total defects above exclude “Cancelled” defects.

5.4.2.2 Conclusion

#	Criteria	Status
1	All the inputs of system testing are baselined and approved by relevant stakeholders (Common requirement, ADD, SRS, Design, Software package, System test cases).	
2	System testing is over and all known issues (if any) are aware by relevant stakeholders (Team lead, Test lead, PM, Tech lead) and documented in Test summary report.	
3	All planned Test Cases have been executed (100% test execution coverage)	
4	At least 95% executed system test cases have to be passed. (More than 95% test successful coverage)	
5	100% high priority test cases in scope are passed.	
6	Only Cosmetic defects can be accepted	

Table 123 - System Test Success Criteria

VI. Chapter VI - Software User's Manual

6.1 Installation Guide

6.1.1 Website

6.1.1.1 Setting Up the Environment at Server Side

- Sign up an account for www.zymmic.com .You can see instruction file “zymic-signup.docx” on folder “Tool&Support/” from disk
- FILEZILLA software version 3.6.0.2. You can get from Disk on folder “Tool&Support”.
- Package for deployment you can get it from disk on folder “Web”
- Create a database on <http://www.zymic.com/> with your host name, username.

And name of database: “yourhostname_sever”

(Our example: “giangvh007_zxq_sever”) and you must see database design for create table, column, and type of data. You can see instruction link:

http://www.youtube.com/watch?v=wJr_H0t9NkA

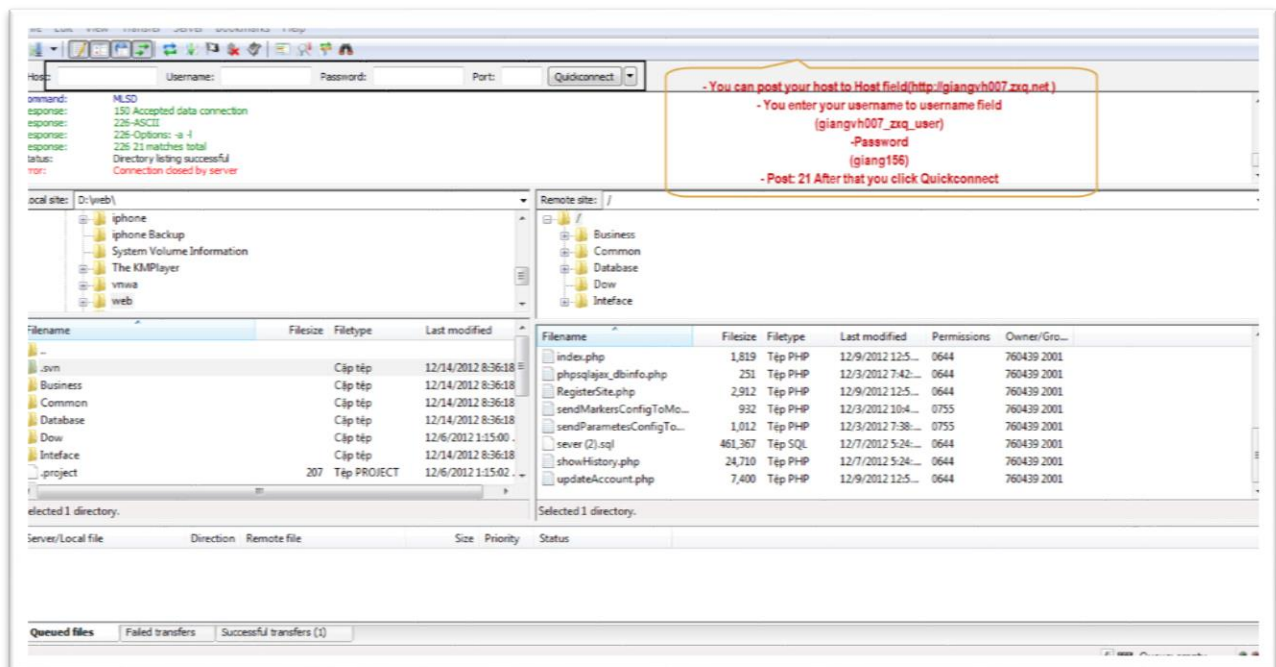
6.1.1.2 Deployment at Server Side

- **Step 1:** Connect to file manager by FileZilla.

Open FileZilla software and

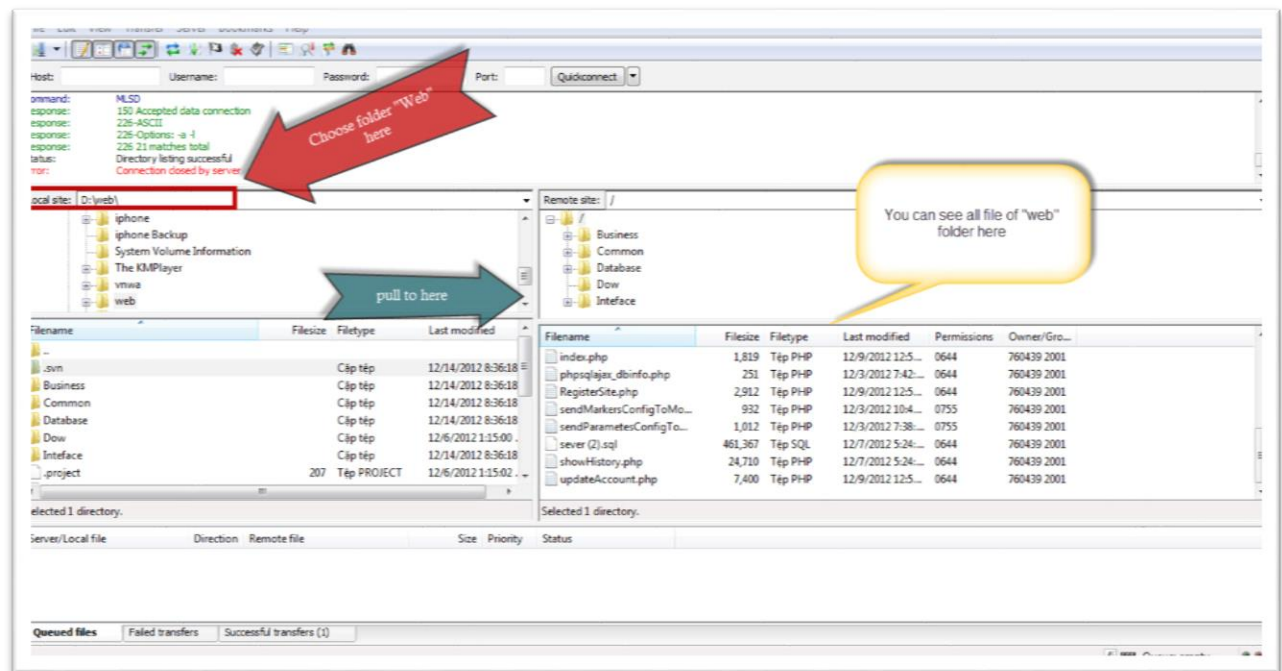
- You can post your host to Host field(fx: <http://giangvh007.zxq.net>)
- You enter your username to username field (fx: [giangvh007_zxq_user](#))
- Password (fx :giang156)
- Post (fx: 21)

After that you click Quickconnect button (You should use your account for connect to host)



- **Step 2:** Transfer Web folder to file manager of zymic by FileZilla.
 - You choose “web” folder at local site field (for example with following image.)

- Pull Web folder to next table after that you can see result



6.1.1.3 Setting Up the Environment at Client Side

The system is accessed via web browsers. The client machine should be installed with one of the following browsers: Firefox 11.0 and higher, Chrome, and Internet Explorer 7.0 and higher.

6.1.2 Application

6.1.2.1 Build and Get apk File

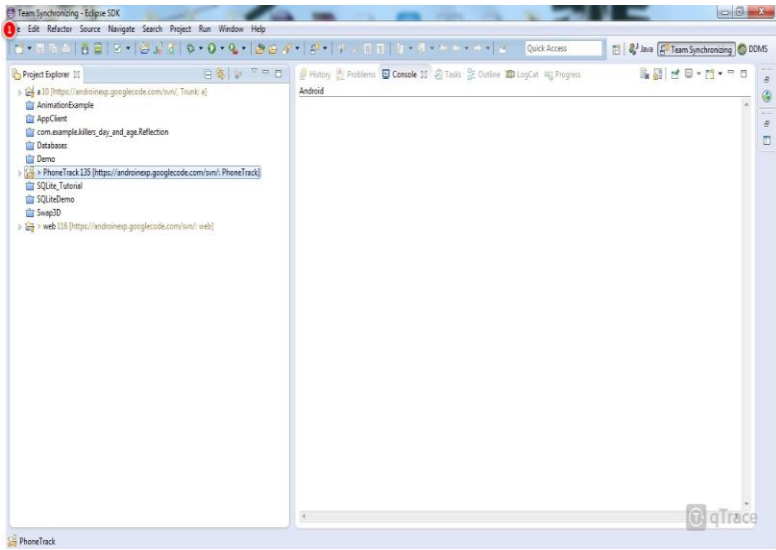
Summary

Description:	
Trace Steps:	<p>Screen 1: Eclipse SDK</p> <ul style="list-style-type: none"> • Step 1: Select 'File' menu item <p>Screen 2: Eclipse SDK</p> <ul style="list-style-type: none"> • Step 2: Select 'Import...' menu item <p>Screen 3: Import</p> <ul style="list-style-type: none"> • Step 3: Double click 'File System' tree item

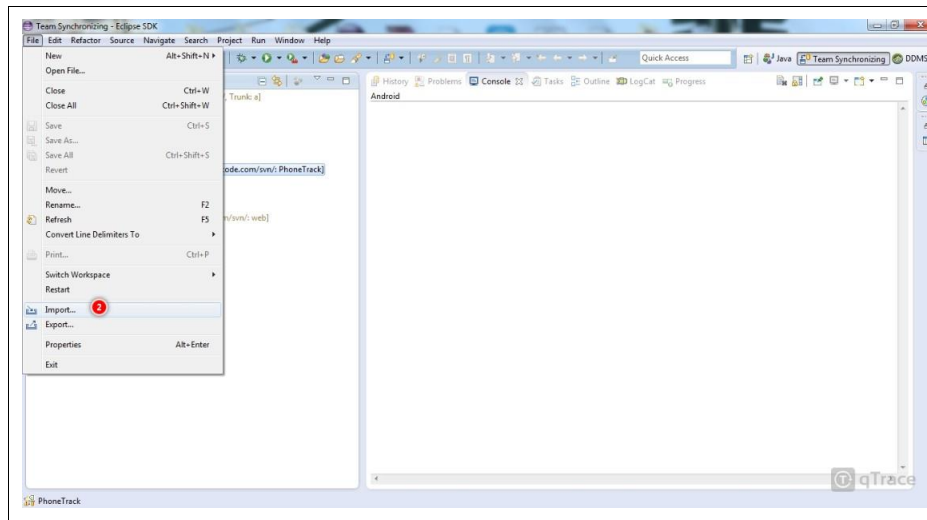
	<p><u>Screen 4:</u> Import</p> <ul style="list-style-type: none"> • Step 4: Click 'Browse...' button <p><u>Screen 5:</u> Import from directory</p> <ul style="list-style-type: none"> • Step 5: Select 'Đĩa Cục bộ (C:)' tree item <p><u>Screen 6:</u> Import from directory</p> <ul style="list-style-type: none"> • Step 6: Click 'Position' button <p><u>Screen 7:</u> Import from directory</p> <ul style="list-style-type: none"> • Step 7: Select 'Người dùng' tree item <p><u>Screen 8:</u> Import from directory</p> <ul style="list-style-type: none"> • Step 8: Select 'giangvh00848' tree item <p><u>Screen 9:</u> Import from directory</p> <ul style="list-style-type: none"> • Step 9: Select 'workspace' tree item <p><u>Screen 10:</u> Import from directory</p> <ul style="list-style-type: none"> • Step 10: Click 'OK' button <p><u>Screen 11:</u> Import</p> <ul style="list-style-type: none"> • Step 11: Click here <p><u>Screen 12:</u> Import</p> <ul style="list-style-type: none"> • Step 12: Click 'Finish' button <p><u>Screen 13:</u> Eclipse SDK</p> <ul style="list-style-type: none"> • Step 13: Right click '> PhoneTrack 135 [https://androidexp.googlecode.com/svn/: PhoneTrack]' tree item <p><u>Screen 14:</u> Team Synchronizing - Eclipse SDK</p> <ul style="list-style-type: none"> • Step 14: Select '1 Android Application' menu item <p><u>Screen 15:</u> Team Synchronizing - Eclipse SDK</p>
Environment:	<p>System Information</p> <ul style="list-style-type: none"> • Microsoft Windows 7 Ultimate Edition Service Pack 1 (build

	<p>7601), 64-bit, Major=6 Minor=1</p> <ul style="list-style-type: none">• qTrace version:2.6.2.1• IE version: 9.0.8112.16421• ADO version: 6.1.7601.17514• CPU: 4 Speed=2.05GHz, Intel64 Family 6 Model 42 Stepping 7, GenuineIntel, Intel(R) Core(TM) i3-2310M CPU @ 2.10GHz• Total physical memory: 4043.86MB• Available physical memory: 2104.12MB• User: TungBach (Admin rights=1) <p>Recorded Application 1: eclipse.exe</p> <ul style="list-style-type: none">• Application version:• File path: E:\eclipse\eclipse.exe• File version:• File size: 0.30MB (319488 Bytes)• File modified date: 06/08/2012 07:59:32
--	--

Screen 1: Eclipse SDK

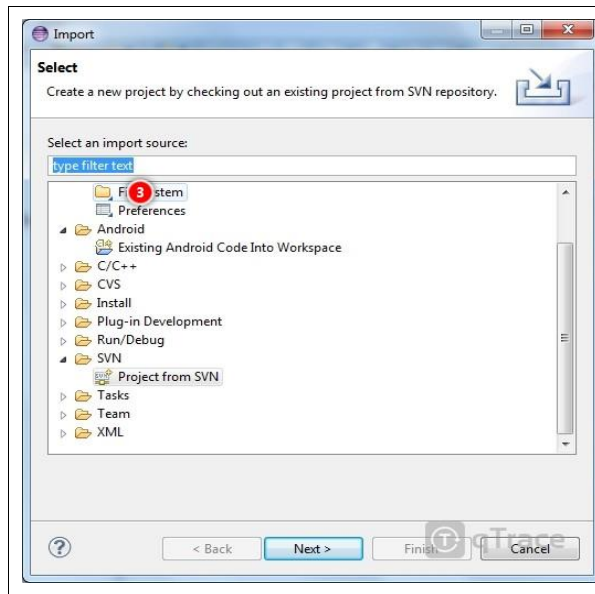
	<p>Step 1: Select 'File' menu item</p>
---	---

Screen 2: Eclipse SDK



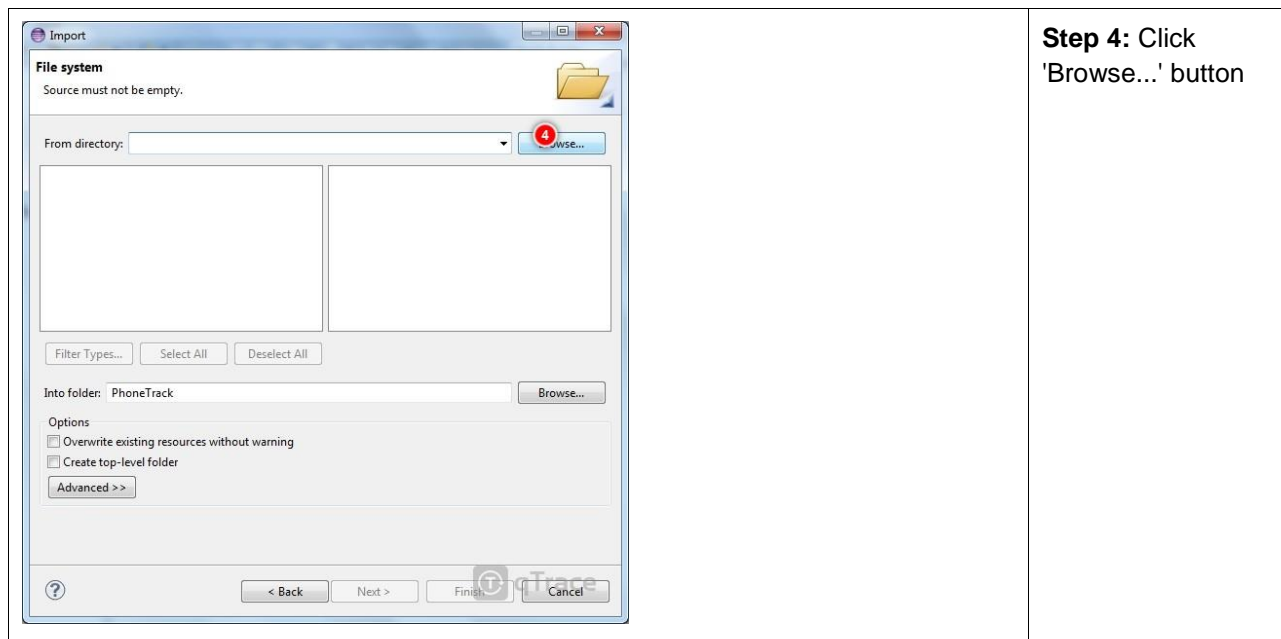
Step 2: Select 'Import...' menu item

Screen 3: Import

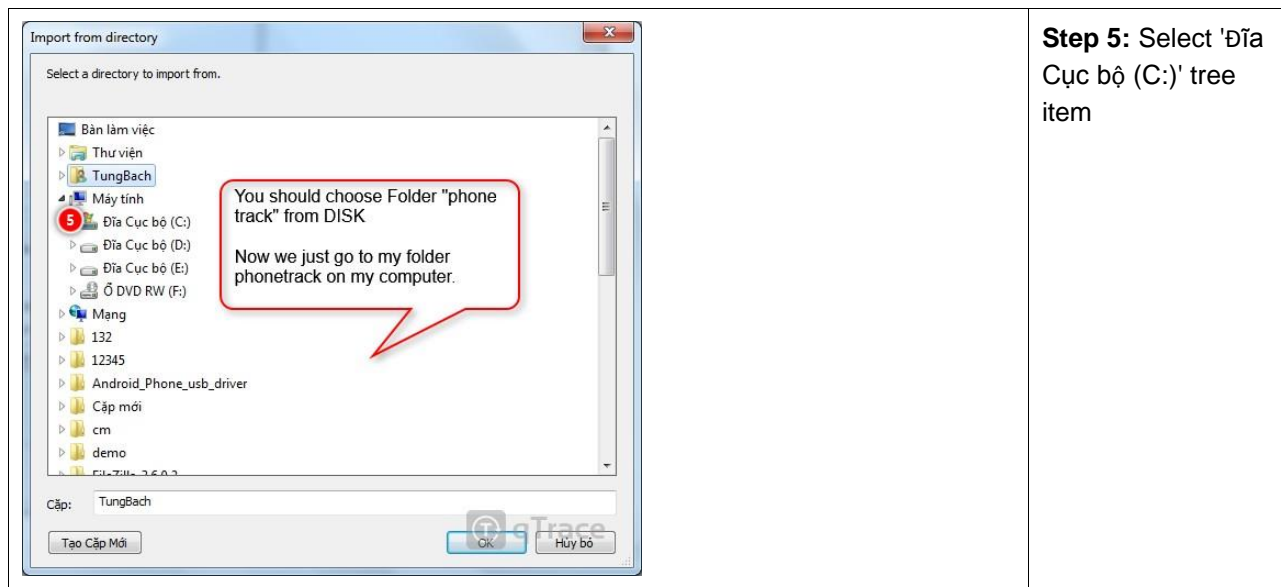


Step 3: Double click 'File System' tree item

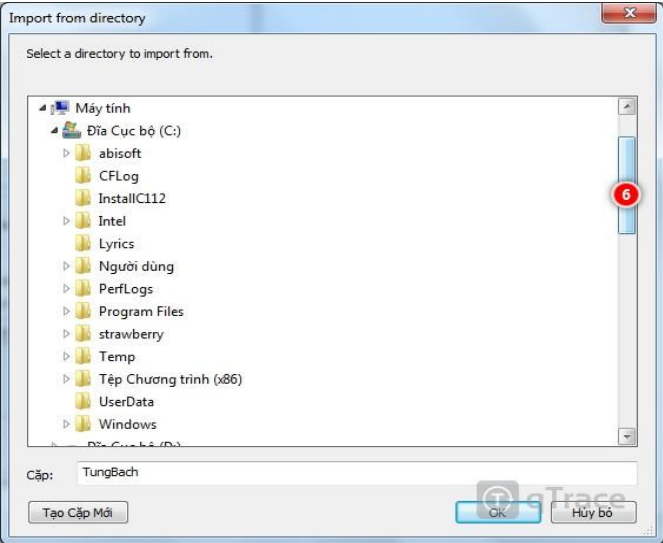
Screen 4: Import



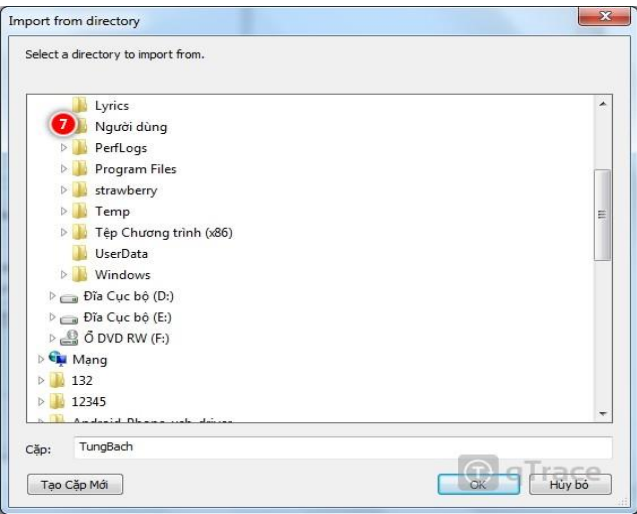
Screen 5: Import from directory



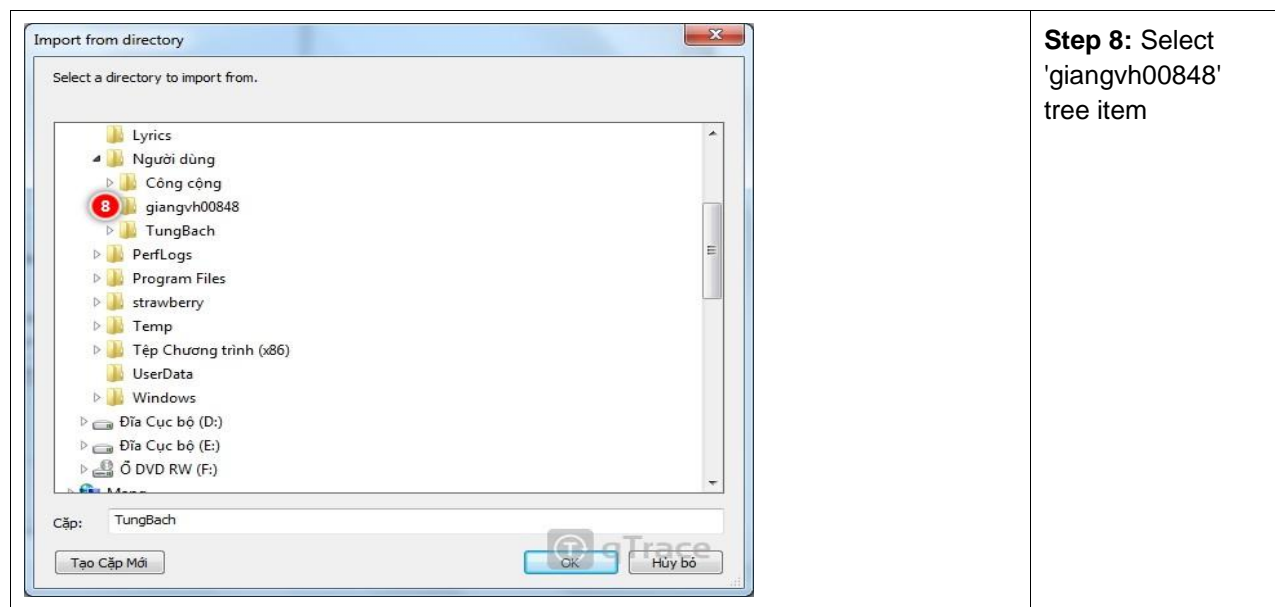
Screen 6: Import from directory

	<p>Step 6: Click 'Position' button</p>
---	---

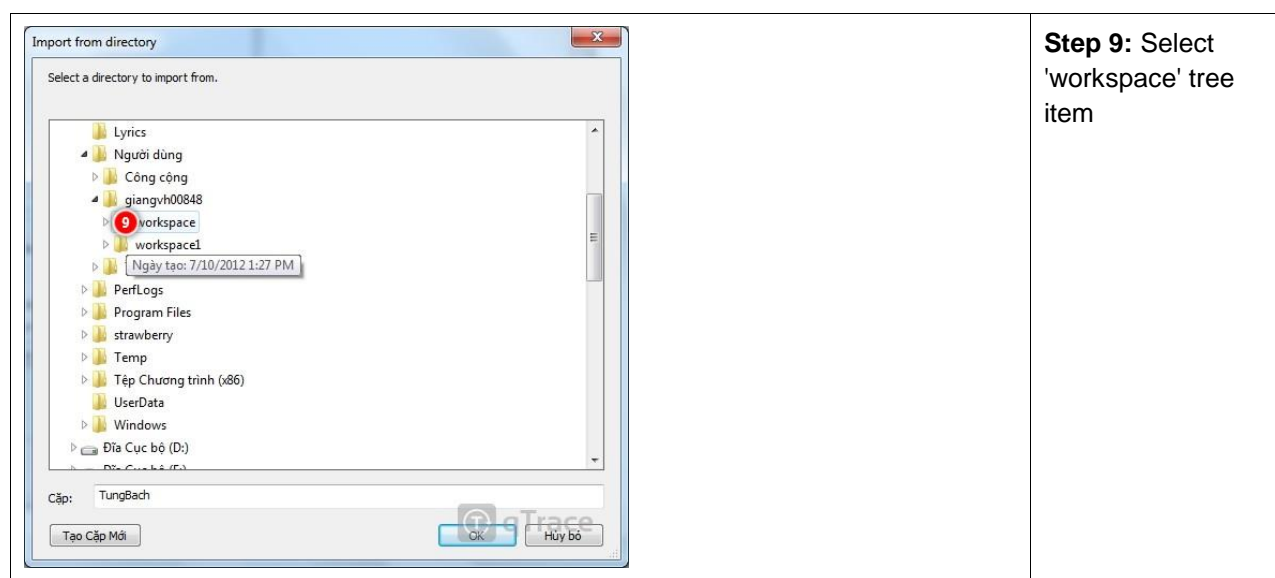
Screen 7: Import from directory

	<p>Step 7: Select 'Người dùng' tree item</p>
--	---

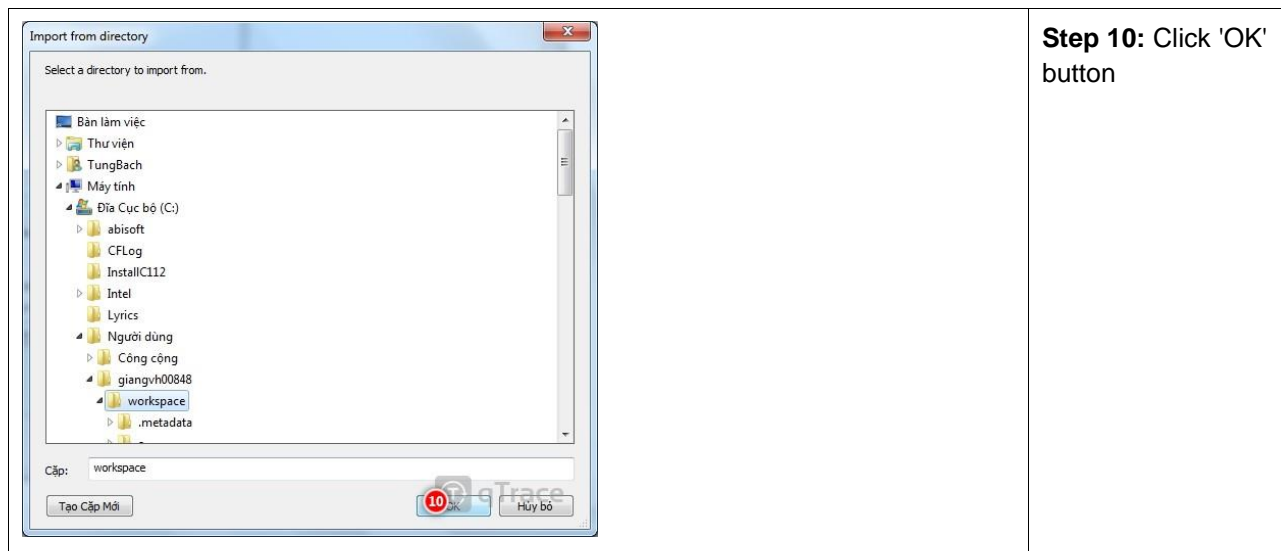
Screen 8: Import from directory



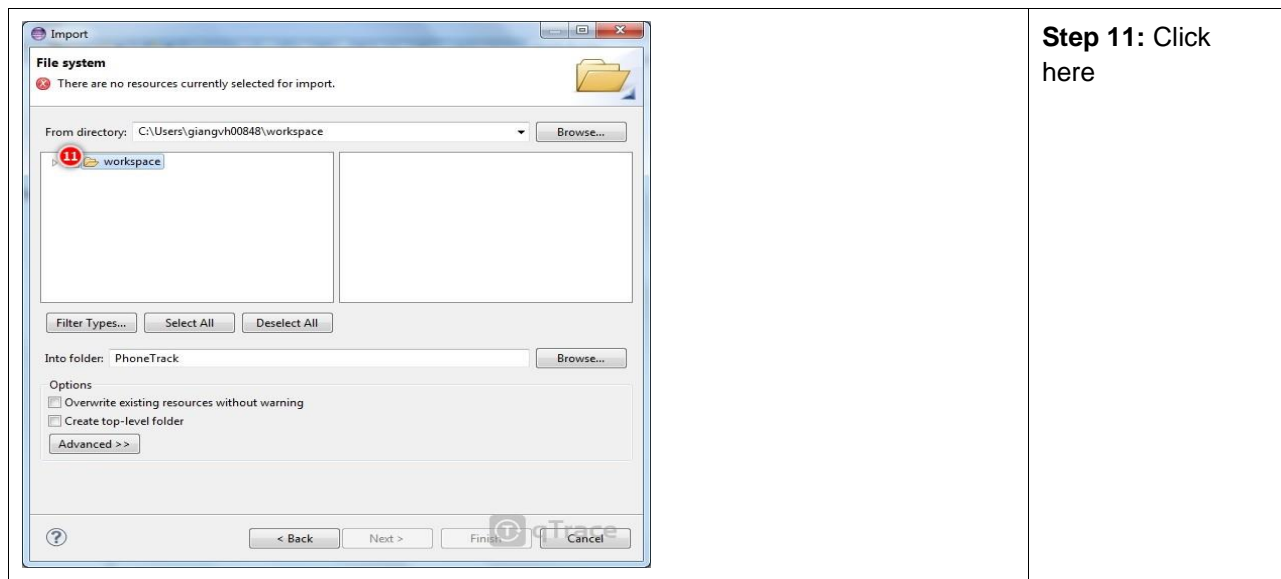
Screen 9: Import from directory



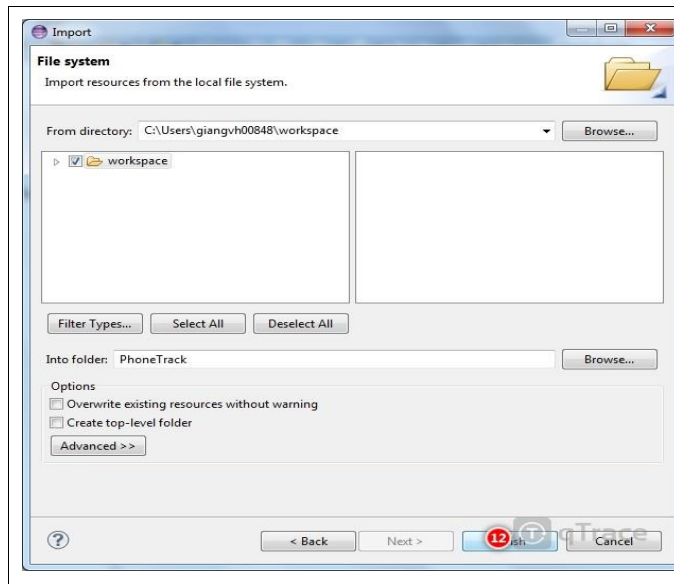
Screen 10: Import from directory



Screen 11: Import

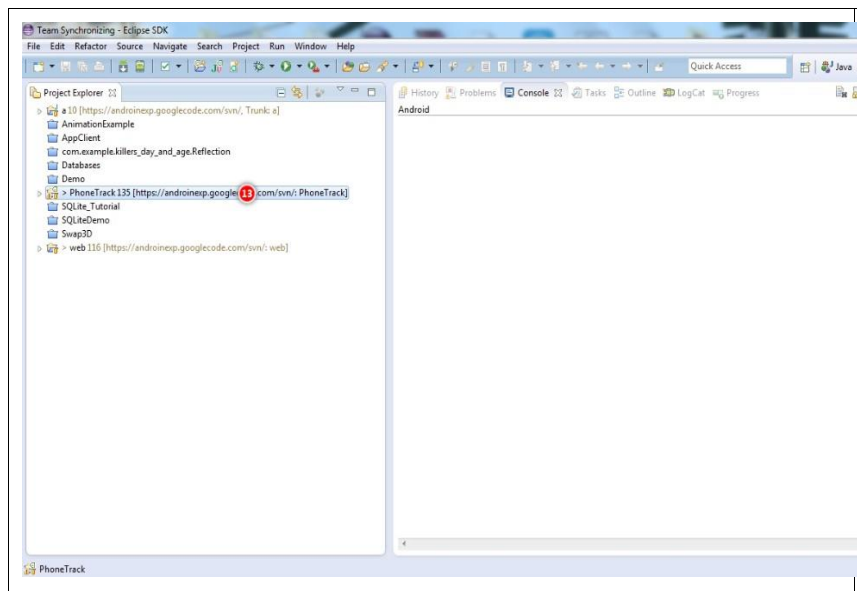


Screen 12: Import



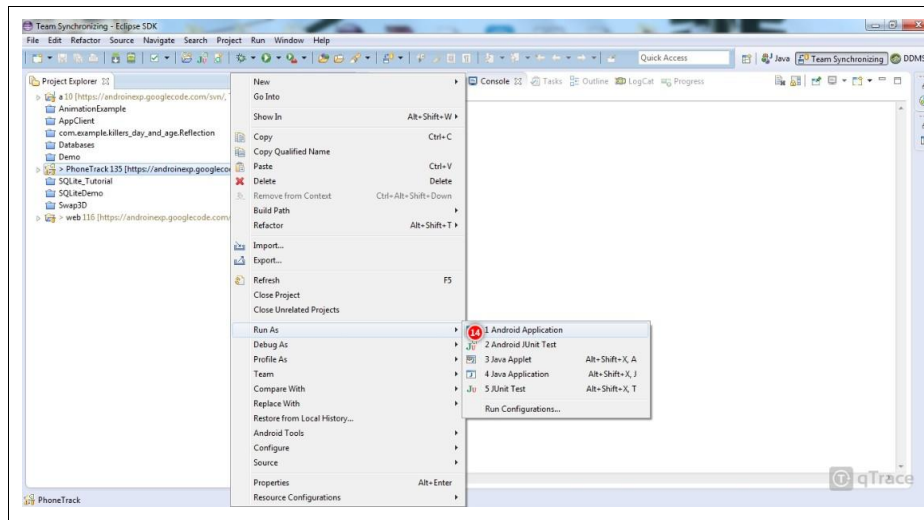
Step 12: Click 'Finish' button

Screen 13: Eclipse SDK



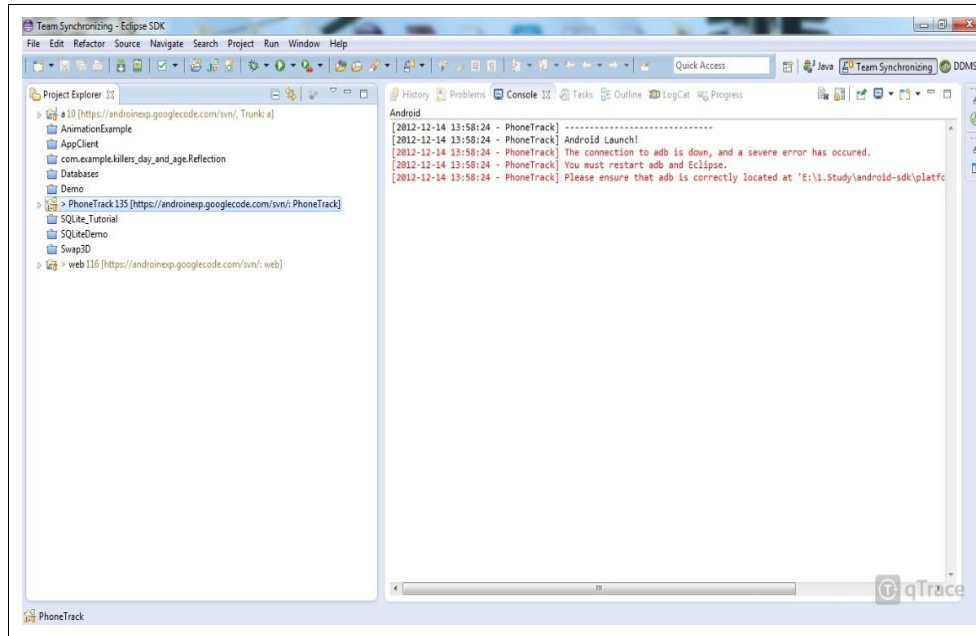
Step 13: Right click '>' PhoneTrack 135 [https://androidexp.googlecode.com/svn/: PhoneTrack]

Screen 14: Team Synchronizing - Eclipse SDK



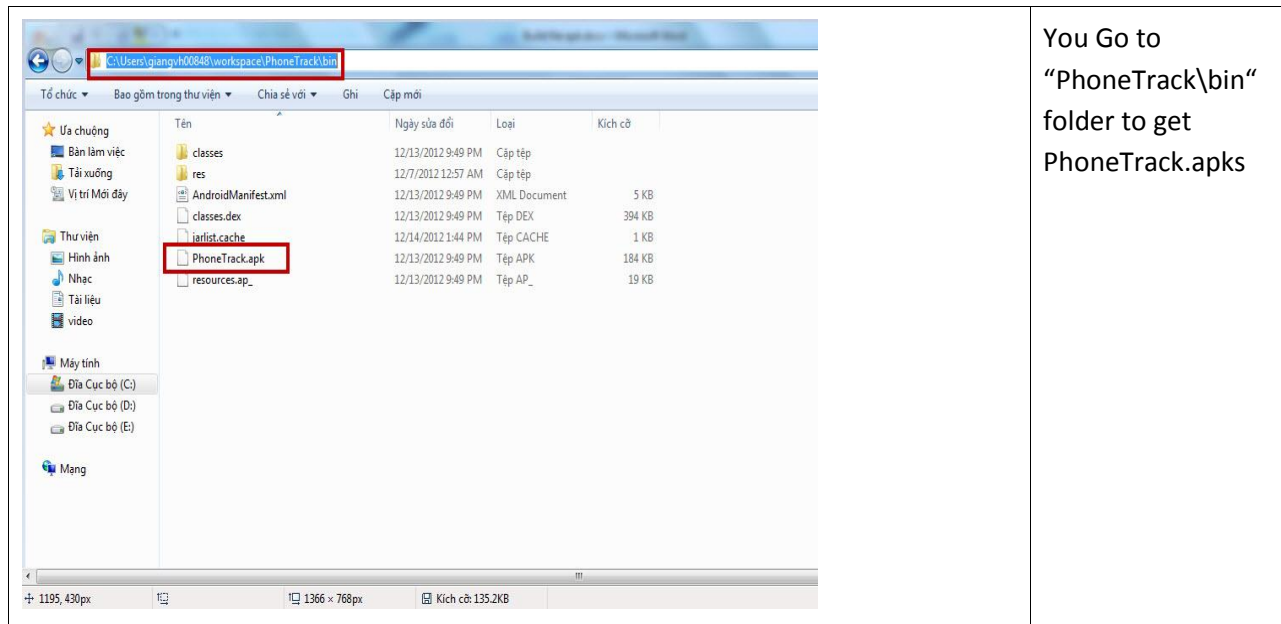
Step 14: Select '1 Android Application' menu item

Screen 15: Team Synchronizing - Eclipse SDK



After Run application you can see the result.

Screen 16 : Go to folder \PhoneTrack\bin you can get phonetrack.apk



6.1.2.2 Set Up Application to Device

6.1.2.2.1 Prepare for Setting

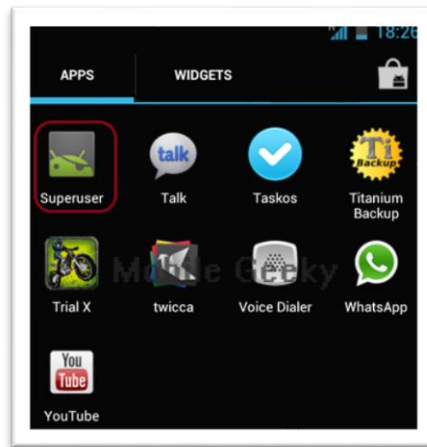
- Android device with version 2.3 or over. (we are testing on Samsung galaxy S2 for this document)
- Odin software (version 1.85). You can down load on "Tool&Support" folder from disk
- ES file explorer or root explorer software (Version 1.6.2.3). You can down load from:
- <https://play.google.com/store/apps/details?id=com.estrong.android.pop&hl=en>

6.1.2.2.2 Set Up Application

- **Step 1:** Root android device. Because there are too much android device so we will take an example: device Samsung galaxy S2 you can see instruction video below:

http://www.youtube.com/watch?v=WYE_gpqCUgc

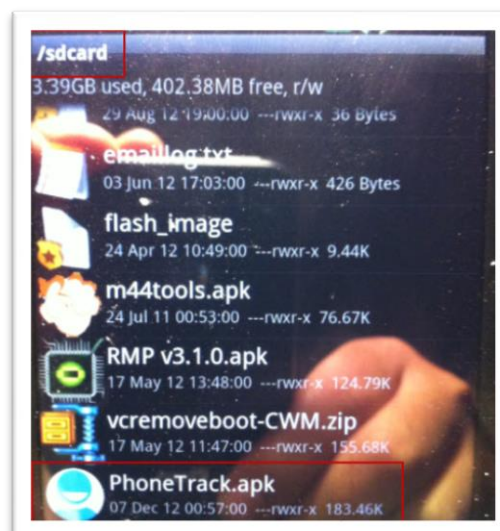
After root device you can see super user icon



- **Step 2:** Install ES file explorer or root explorer, after that you can see below icon.



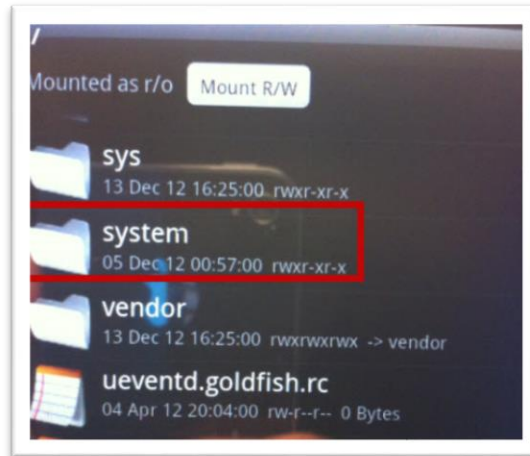
- **Step 3:** Get phonetrack.apk to device from Disk to SD card or internal memory. For example: /sdcard/phonetrack.apk



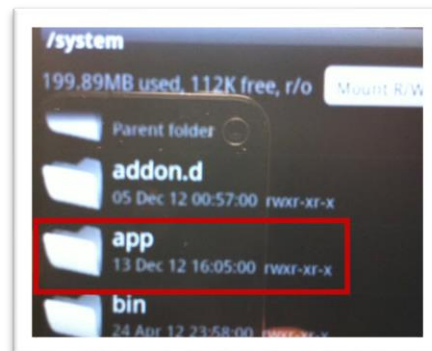
- **Step 4:** Open Root explorer or ES file explorer and move phonetrack.apk to folder: system/app.

Click on Root explorer Icon and choose on parent folder.

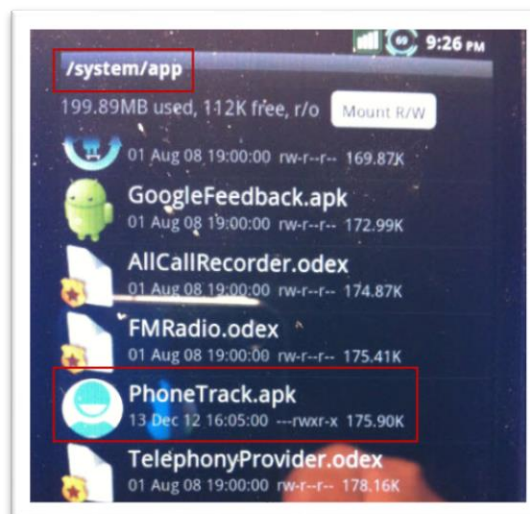
Choose system folder



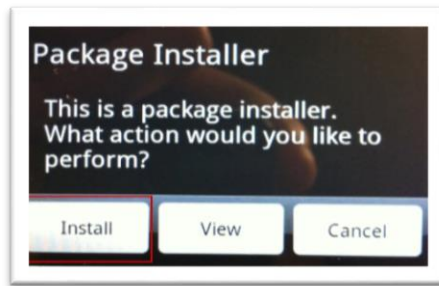
Choose app folder.



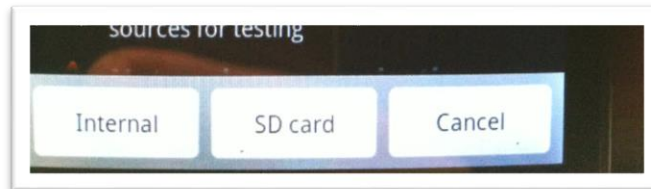
Paste phonetrack.apk file here.



Step 5: Install application by click on phonetrack.apk Icon, after that message box will showed



After you click install button you can see



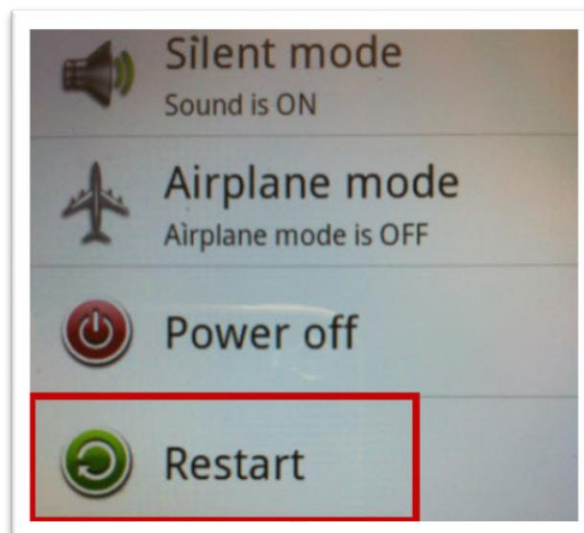
You click internal button.



Click done button.

- **Step 6:** Reset device.

For example with galaxy s2 you can press and hold power button what is up on device, the box will showed. Choose restart option.

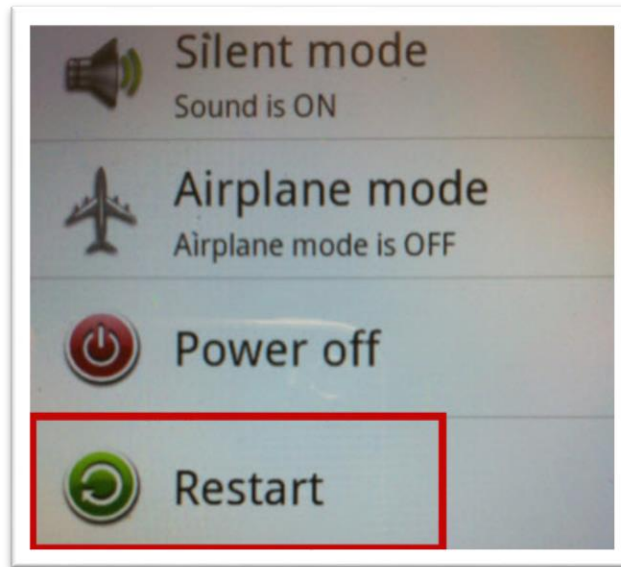


- **Step 7:** UNROOT device. Specification with device Samsung galaxy S2 you can see instruction video below:

<http://www.youtube.com/watch?v=tW6x29ug5Ms>.

- **Step 8:** Reset device.

For example with galaxy s2 you can press and hold power button what is up on device, the box will showed. Choose restart option.



6.2 User's Guide

6.2.1 General Guide

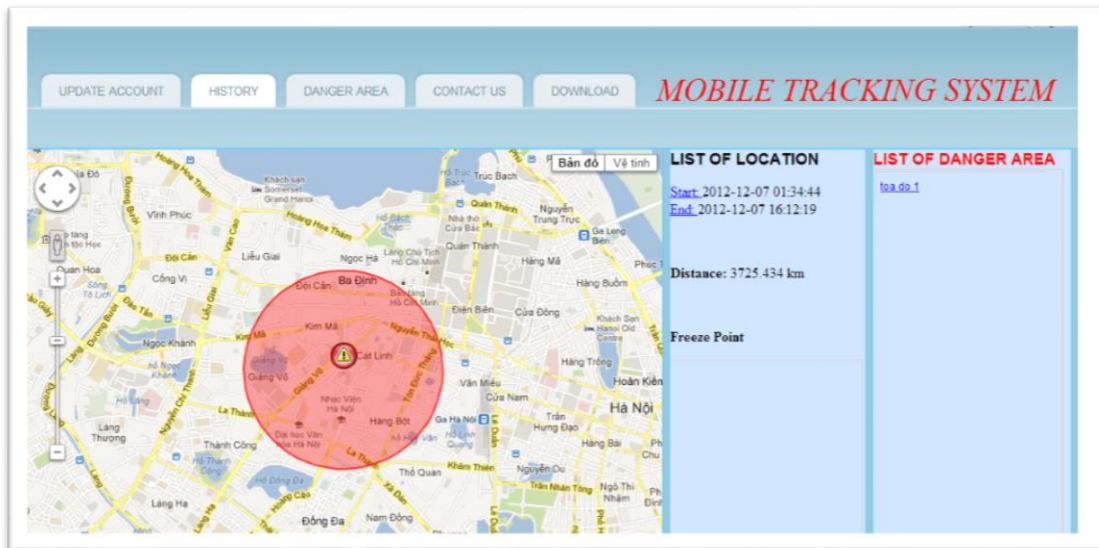
6.2.1.1 Login

- **Step 1:** Type in 'Địa chỉ' edit box: 'giangvh007.zxq.net'
- **Step 2:** Press Enter



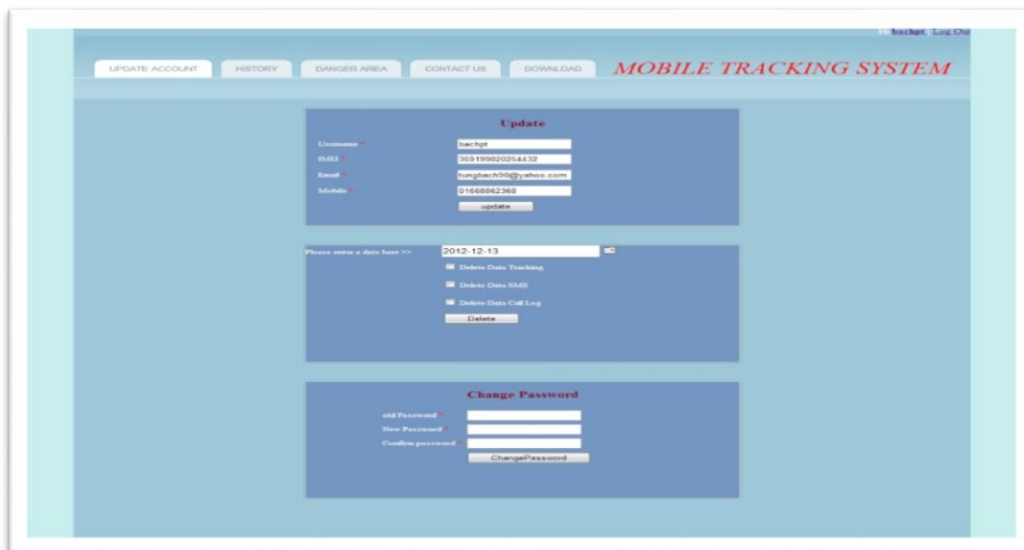
- **Step 3:** Type in 'Username:' edit box:

- **Step 4:** Type in 'Password:' edit box:
- **Step 5:** Click 'Sign in' button and History screen will be showed



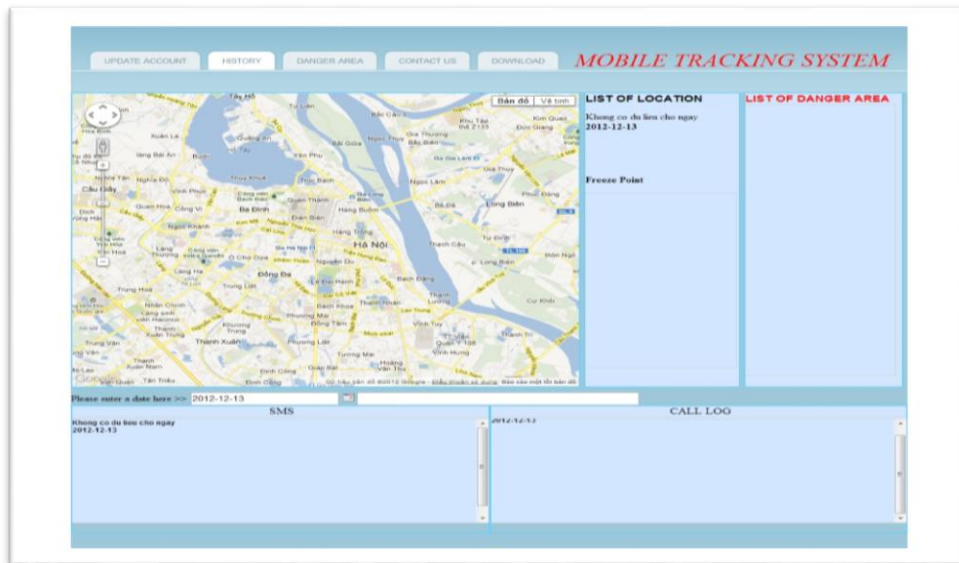
6.2.1.2 View Profile

- **Step 1:** Login and Click Update Account Tab Update screen will be showed



6.2.1.3 View Position, SMS, Call Logs of Current Day

- **Step 1:** Login and History screen will be showed



6.2.1.4 View Danger Area Screen

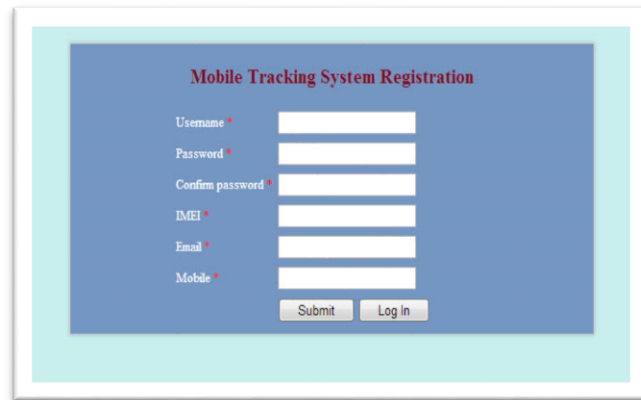
- **Step 1:** Login and Click Danger Area Tab and Danger Area screen will be showed

6.2.2 Form Guide

6.2.2.1 Register a New Account

- **Step 1:** Go to login Screen and click on “Create an account”

- **Step 2:** After Register was showed, you need to enter all information
 - User name field, password, confirm password must be from 6 to 12 characters.
 - IMEI must be exactly 15 characters
 - EMAIL must be right form of email.
 - Number phone just is a number



Mobile Tracking System Registration

Username *

Password *

Confirm password *


IMEI *

Email *

Mobile *

6.2.2.2 Forgot Password

- **Step 1:** Go to login Screen and click on “Forgot password”



Username:

Password:

[Create an account](#) | [Forgot password?](#)

click here

6.2.2.3 User's Update Information

- **Step 1:** Login and move to update account tab.
- **Step 2:** You can update IMEI, phone number, email after that you click update button



Hi [kyenonline](#) | [Log Out](#)

[UPDATE ACCOUNT](#) [HISTORY](#) [DANGER AREA](#) [CONTACT US](#) [DOWNLOAD](#) **MOBILE TRACKING SYSTEM**

Update

Username *

IMEI *

Email *

Mobile *

6.2.2.4 Delete Tracking, SMS and Call Logs Data

- **Step 1:** Login and move to update account tab.
- **Step 2:** You click on calendar icon choose a day and
- **Step 3:** Click on radio button of Track or SMS or Call Logs or all three radio button.
- **Step4:** Click delete button

The screenshot shows a web interface titled "Update" with a blue background. It contains several input fields for account information: Username (bachpt), IMEI (359199020254432), Email (tungbach90@yahoo.com), and Mobile (1444). To the right of these fields is a calendar for December 2012. A yellow callout bubble points to the calendar with the text "Choose a day". Below the calendar, there is a date input field showing "2012-12-07". Underneath the date field are three radio button options: "Delete Data Tracking", "Delete Data SMS", and "Delete Data Call Log". A yellow callout bubble points to these options with the text "choose what you want to delete". At the bottom of these options is a "Delete" button. A yellow callout bubble points to this button with the text "click delete button".

Mo	Tu	We	Th	Fr	Sa	Su
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

6.2.2.5 Change Password

- **Step 1:** Login and move to update account tab.
- **Step 2:** You enter your current password to old password field
- **Step 3:** you enter new password to new password, and confirm password field
- **Step4:** click change button

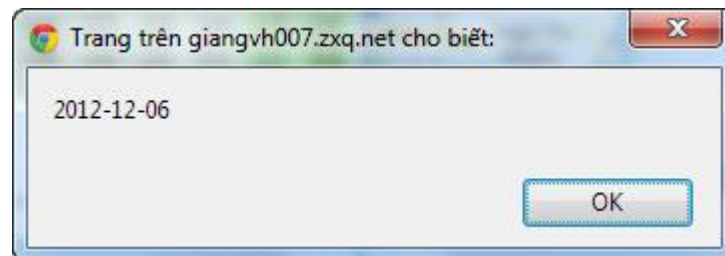
6.2.2.6 View History of Position, SMS, Call Logs

- **Step 1:** Login
- **Step 2:** Choose a day on calendar icon

ID	Type	Phone Number	Time	
153	incoming	7666	2012-12-06 14:34:59	hi em
154	outgoing	7666	2012-12-06 14:35:22	Sao
155	incoming	7666	2012-12-06 14:36:30	ko

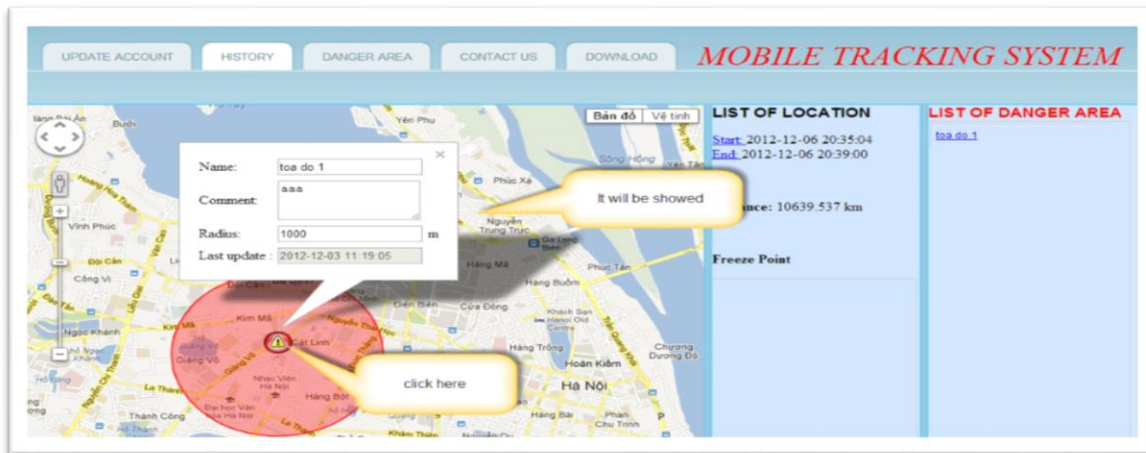
ID	Type	Phone Number	Time	Duration
310	Receive Call	123	2012-12-06 19:29:14	1
312	Miss Call	6666	2012-12-06 20:23:45	
313	Miss Call	8888	2012-12-06 20:23:57	
314	Miss Call	8888	2012-12-06 20:36:09	
315	Receive Call	1111	2012-12-06 20:36:18	1

- **Step 3:** click ok button for result.



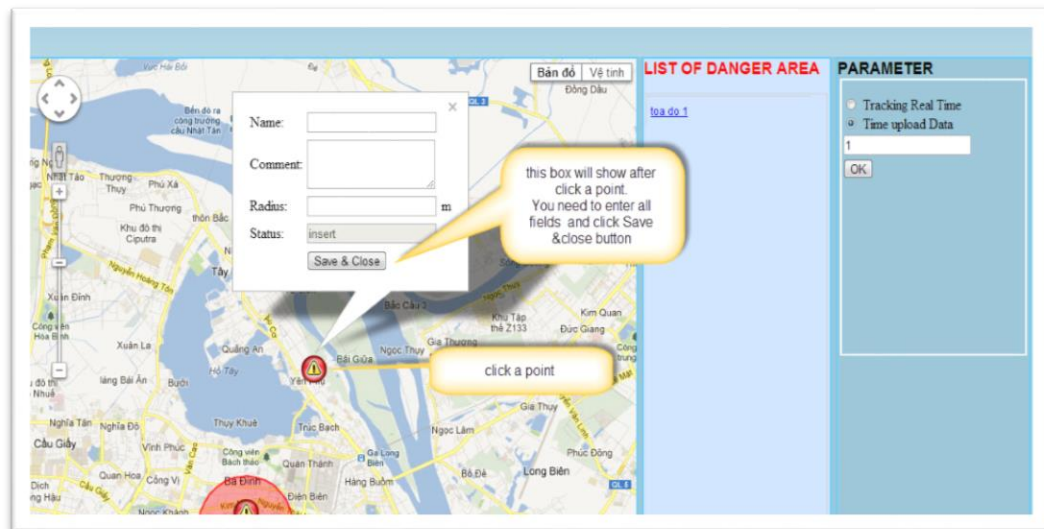
6.2.2.7 View Information of Danger Point

- **Step 1:** Login
- **Step 2:** Click on a danger point like below image.



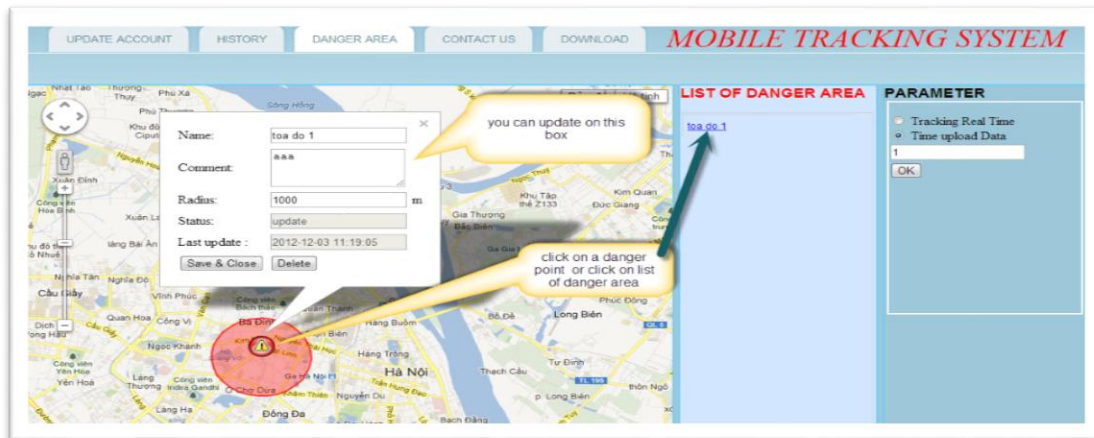
6.2.2.8 Create a New Danger Point

- **Step 1:** Login and move to danger area tab.
- **Step 2:** You click on a point on map
- **Step 3:** After click on a point information box will be showed, you need to enter all data for this box.
- **Step4:** Click “Save & close” button



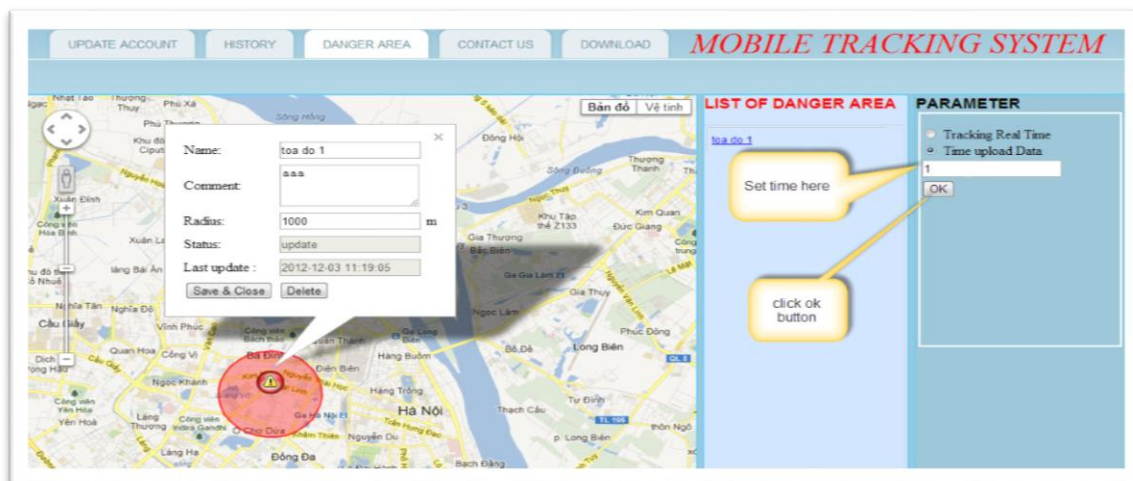
6.2.2.9 Update Information or Delete Danger Point

- **Step 1:** Login and move to danger area tab.
- **Step 2:** You click on a danger point on map
- **Step 3:** After click on a danger point information box will be showed, you can change data on this box.
- **Step4:** Click “Save&close” button or delete button if you want to delete this point

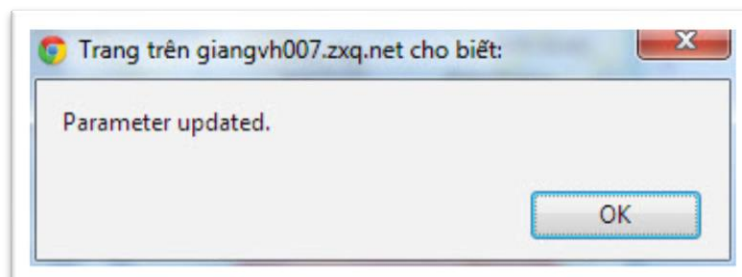


6.2.2.10 Set Time Upload Data

- **Step 1:** Login and move to danger area tab.
- **Step 2:** You set a time on time upload textbox
- **Step 3:** click ok button, a message box will be showed



- **Step 4:** Click ok



6.2.3 Admin Guide

6.2.3.1 View All Information of Users

- **Step 1:** Login with admin account after that admin screen will be showed. You can see all information of users: username, password, role, IMEI, phone number, email.

MOBILE TRACKING SYSTEM						
ID	User Name	Password	Role	Imei	Phone Number	Email
7	giangvh001	e10adc3949ba59abbe56e057f20f883e	2	111111111111111	0901111111111	giangvh001@yahoo.com.vn
11	admin1234	e10adc3949ba59abbe56e057f20f883e	1	999999999999999	09099999999	admin1234@yahoo.com.vn
4	nguyeniden	e10adc3949ba59abbe56e057f20f883e	2	358490041283939	0912862752	white1990black@yahoo.com
5	kyenonline	e10adc3949ba59abbe56e057f20f883e	2	000000000000000	0912862751	white1990black@yahoo.com
8	giangvh002	a5d4b4c9fb4308176bcb9e72a2b674b	2	222222222222222	09022222222	giangvh002@yahoo.com.vn

6.2.3.2 Update Information of Users or Delete Users

- **Step 1:** Login with admin account after that admin screen will be showed.
- **Step 2:** click on a row after that information box will be showed

Account Info

Id	5
User Name	kyenonline
Role	User
IMEI	Admin Locked 000000
Mobile	0912862751
Email	white1990black@yahoo.com

update

delete

resetPassword

after you change click update for change

you can change role, imei, mobile, email

you can click delete for delete user, or reset password for reset password for users

Account Info

Id	19
User Name	bachpt
Role	User
IMEI	359199020254432
Mobile	01668862368
Email	bachpt01389@fpt.edu.vn

update

delete

resetPassword

- **Step 3:** Admin can change role, imei, phone number, email

6.2.4 User Enables Update For Device by Sending Message

- **Step 1:** After user Register, and set up application successfully on device, Send a message with special format (admin will give you) to number of phone what has IMEI registered.

VII. References

1. <https://play.google.com/store/apps>
2. http://en.wikipedia.org/wiki/Google_Maps
3. https://play.google.com/store/apps/details?id=com.kendroid.android.routetracker&feature=search_result#?t=W251bGwsMSwxLDEslmNvbS5rZW5kcm9pZC5hb mRyb2lkLnJvdXRldHJhY2tldiJd
4. <https://play.google.com/store/apps/details?id=com.fsp.android.c&hl=vi>
5. <https://play.google.com/store/apps/details?id=com.gizmoquip.smstracker&hl=vi>
6. http://www.youtube.com/watch?v=wJr_H0t9NkA
7. http://www.youtube.com/watch?v=WYE_gpqCUgc
8. <http://www.youtube.com/watch?v=tW6x29ug5Ms>.
9. <https://play.google.com/store/apps/details?id=com.estrong.s.android.pop&hl=en>